

**IMPLEMENTASI *SERVICE ORIENTED*
ARCHITECTURE (SOA) MENGGUNAKAN
TEKNOLOGI *WEB SERVICE* PADA APLIKASI
CUSTOMER SERVICE
(STUDI KASUS: DINAS PENDAPATAN DAERAH DAN
PT. JASARAHARJA (PERSERO) CABANG RIAU)**

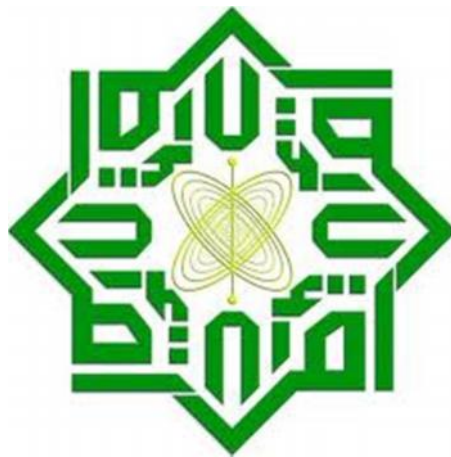
TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik
Pada Jurusan Teknik Informatika

Oleh :

FERDIANSYAH

10951007959



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU**

2013

**IMPLEMENTASI *SERVICE ORIENTED ARCHITECTURE*
(SOA) MENGGUNAKAN TEKNOLOGI *WEB SERVICE* PADA
APLIKASI *CUSTOMER SERVICE***

**(STUDI KASUS: DINAS PENDAPATAN DAERAH DAN PT.
JASARAHARJA (PERSERO) CABANG RIAU)**

FERDIANSYAH

10951007959

Tanggal sidang : 31 Oktober 2013

Periode wisuda : November 2013

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

Jl. Soebrantas KM 15 No. 155 Pekanbaru

ABSTRAK

Service Oriented Architecture (SOA) adalah sebuah layanan *web service* yang dapat mengelola sumber daya dari berbagai informasi yang ada dalam suatu proses bisnis dari banyak perusahaan atau instansi terkait secara efektif dan efisien. Dewasa ini, banyak perusahaan atau instansi yang menghadapi tantangan dalam memenuhi permintaan *customer* dengan layanan yang tinggi, sehingga perusahaan tersebut mengalami kesulitan, salah satunya adalah PT. Jasaraharja (Persero). Dalam penelitian ini diimplementasikan teknologi SOA untuk menyelesaikan permasalahan yang ada. Banyak teknologi yang dapat digunakan untuk mengimplementasikan arsitektur SOA, diantaranya adalah CORBA, DCOM, RMI, dan *Web Service*. Teknologi *Web Service* bersifat terbuka (*non-proprietary*) sehingga teknologi kedepan yang banyak berkembang adalah *web service*. Perancangan *web service* mengikuti standar dari SOA, menggunakan protokol komunikasi SOAP (*simple object access protocol*), WSDL (*web service description language*), serta UDDI (*universal description discovery integration*). Hasil penelitian menyimpulkan bahwa dengan mengimplementasikan teknologi SOA menggunakan *web service* pada PT. Jasaraharja dapat membantu dalam mengelola data secara lebih efisien dari segi waktu dan biaya pengembangan.

Kata Kunci: *SOA, Proprietary, Web Service.*

IMPLEMENTATION OF SERVICE ORIENTED ARCHITECTURE (SOA) IN TECHNOLOGY WEB SERVICE ON CUSTOMER SERVICE APPLICATION

**(CASE STUDY: DINAS PENDAPATAN DAERAH DAN PT.
JASARAHARJA BRANCH RIAU)**

FERDIANSYAH

10951007959

Date Of Final Exam : 31th October 2013

Graduation Ceremony Period : November 2013

Informatic Engineering Department

Faculty Sains And Technology

State Islamic University Of Sultan Syarif Kasim Riau

Jl. Soebrantas KM 15 No. 155 Pekanbaru

ABSTRACT

*Service Oriented Architecture (SOA) is a web service that can organize resources from many kinds of information in a business process from that instantion effectively and efficiently. In this era, many company or instantion which find **challenge** in fulfil customer request highly service, then that company meet difficulties, one of them is PT. Jasaraharja (Persero). In this research was implemented SOA technology to solve that problem. Many technologies that can use to implement SOA architecture, they are CORBA, DCOM, RMI, and web service. Technologies Web Service non proprietary so that many developing future technologies is a web service. Web service followed the standards of SOA, use the communication protocol SOAP (simple object access protocol), WSDL (web service description language), and UDDI (universal description discovery integration). The result of this research conclude that by using implementation of SOA technology used web service at PT. Jasaraharja can Organize date more affiently from the side of times and costs.*

Key words: *SOA, Proprietary, Web Service.*

KATA PENGANTAR

Alhamdulillah Robbil'alamin, penulis ucapkan syukur yang setinggi-tinggi ke-hadirat Allah SWT, karena atas segala limpahan rahmat dan karuniahnya yang diberikan sehingga penulis dapat menyelesaikan penelitian sekaligus penulisan laporan tugas akhir ini. *Allahumma sholli'ala Muhammad wa'ala ali sayyidina Muhammad*, yang tidak lupa penulis haturkan juga untuk junjungan alam, kekasih Allah, Rasul Allah, dan tauladan kita yakni Nabi Muhammad SAW.

Laporan tugas akhir ini merupakan salah satu prasyarat untuk memenuhi persyaratan akademis dalam rangka meraih gelar kesarjanaan di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau (UIN SUSKA Riau). Selama menyelesaikan tugas akhir ini, penulis telah banyak mendapatkan bantuan, bimbingan, dan petunjuk dari banyak pihak baik secara langsung maupun tidak langsung. Untuk itu dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Dra. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Elin Haerani, S.T, M.Kom, selaku Ketua Jurusan Teknik Informatika, Fakultas Sains dan Teknologi.
4. Iwan Iskandar, S.T, M.T Selaku dosen pembimbing I tugas akhir. Terimakasih banyak pak atas dukungan, ilmu serta semangat yang Bapak berikan dalam membimbing saya mengerjakan Tugas Akhir ini.
5. Muhammad Affandes, S.T, M.T sebagai koordinator tugas akhir yang telah memberi masukan-masukan untuk penyelesaian tugas akhir ini, dan sangat sabar membantu penulis dalam mempersiapkan semua kebutuhan penulis dalam penyelesaian Tugas Akhir ini.
6. Jasril, S.Si, M.Sc selaku dosen penguji 1, terimakasih pak untuk ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukannya untuk penyempurnaan laporan ini, terimakasih pak.

7. Muhammad Affandes, ST, MT selaku dosen penguji 2, terimakasih pak untuk ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukannya untuk penyempurnaan laporan ini, terimakasih pak.
 8. Bapak, Ibu, dan Adik-adik Penulis yang selalu memberikan doa, dorongan, semangat untuk penulis.
 9. Terima kasih kepada M. Yunus Syahrin atas saran, waktu dan kesempatan yang diberikan kepada penulis.
 10. Terima kasih kepada Weli, Erzi yang memberi semangat dan selalu siap jika penulis sedang dalam butuh bantuan. Terima kasih Bang.
 11. Terima kasih kepada teman-teman TIF A 09, JPC yang tidak bisa disebutkan satu per satu yang selalu memberi senyum dan semangat yang luar biasa.
 12. Dan terakhir, terimakasih pula penulis ucapkan untuk Almamater Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau serta pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu. Terimakasih banyak atas bantuan dan dukungannya yang berharga.
- Akhirnya, penulis menyadari dalam penulisan laporan ini masih terdapat kekurangan. Oleh karena itu, saran dan kritik sangat penulis harapkan untuk kemajuan penulis secara pribadi. Terimakasih.

Pekanbaru, Oktober 2013

Penulis

DAFTAR ISI

| | |
|--|-------|
| HALAMAN JUDUL LAPORAN..... | i |
| LEMBAR PERSETUJUAN..... | ii |
| LEMBAR PENGESAHAN | iii |
| LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL..... | iv |
| LEMBAR PERNYATAAN | v |
| LEMBAR PERSEMBAHAN | vi |
| ABSTRAK | vii |
| <i>ABSTRACT</i> | viii |
| KATA PENGANTAR | ix |
| DAFTAR ISI | xi |
| DAFTAR GAMBAR | xiv |
| DAFTAR TABEL..... | xvi |
| DAFTAR LAMPIRAN..... | xviii |
| DAFTAR SIMBOL..... | xix |
| BAB I PENDAHULUAN | I-1 |
| 1.1. Latar Belakang | I-1 |
| 1.2. Rumusan Masalah | I-3 |
| 1.3. Batasan Masalah | I-3 |
| 1.4. Tujuan Penelitian..... | I-3 |
| 1.5. Sistematika Penulisan | I-3 |
| BAB II LANDASAN TEORI | II-1 |
| 2.1. <i>Service Oriented</i> | II-1 |
| 2.2. <i>Service Oriented Architecture</i> (SOA)..... | II-2 |
| 2.3. Konsep <i>Service Oriented Architecture</i> (SOA) | II-3 |
| 2.4. Prinsip - Prinsip <i>Service Oriented Architecture</i> (SOA)..... | II-5 |
| 2.5. <i>Web Service</i> | II-8 |
| 2.6. <i>Service Oriented Architecture</i> (SOA) dan <i>Web Service</i> | II-9 |
| 2.7. <i>Unified Modelling Language</i> (UML) | II-10 |
| 2.7.1. <i>Usecase Diagram</i> | II-10 |
| 2.7.2. <i>Process Diagram</i> | II-11 |

| | |
|--|-------|
| 2.8. <i>Customer Service</i> | II-11 |
| 2.9. PT. Jasaraharja (Persero) | II-12 |
| 2.9.1. Struktur Organisasi | II-12 |
| 2.9.2. Dasar Hukum | II-12 |
| 2.9.3. Fungsi PT. Jasaraharja (Persero)..... | II-13 |
| 2.9.4. Kegiatan PT. Jasaraharja (Persero) | II-13 |
| 2.10. Tugas Pokok..... | II-15 |
| 2.11. Kerjasama PT. Jasaraharja (Persero)..... | II-15 |
| 2.11.1. Dengan Pemerintah Daerah Kabupaten dan Kota... | II-15 |
| 2.11.2. Kerjasama dengan Polri | II-16 |
| BAB III METODOLOGI PENELITIAN | III-1 |
| 3.1. Persiapan Penelitian | III-2 |
| 3.1.1. Penelitian Pendahuluan | III-2 |
| 3.1.2. Identifikasi Masalah | III-2 |
| 3.2. Tahapan Penelitian | III-2 |
| 3.2.1. Jadwal Penelitian..... | III-2 |
| 3.3. Pengumpulan Data | III-3 |
| 3.3.1. Studi Literatur | III-3 |
| 3.3.2. Wawancara..... | III-3 |
| 3.4. Analisa dan Perancangan Sistem | III-3 |
| 3.5. Implementasi dan Pengujian | III-4 |
| 3.6. Penarikan Kesimpulan | III-4 |
| BAB IV ANALISA DAN PERANCANGAN | IV-1 |
| 4.1. Analisa..... | IV-1 |
| 4.1.1. Analisa Sistem Lama..... | IV-1 |
| 4.1.1.1. Hasil Wawancara Dispenda | IV-2 |
| 4.1.1.2. Hasil Wawancara Jasaraharja..... | IV-5 |
| 4.1.1.3. Bizagi Dispenda | IV-7 |
| 4.1.1.4. Bizagi Jasaraharja..... | IV-10 |
| 4.1.2. Analisa Sistem Baru | IV-30 |
| 4.1.2.1. <i>Analisa Service-Oriented</i> | IV-30 |
| 4.1.2.2. Dekomposisi Proses Bisnis | IV-35 |

| | | |
|----------------|--|-------|
| 4.2. | Perancangan Aplikasi..... | IV-36 |
| 4.2.1. | Perancangan UML (<i>Unified Modeling Language</i>) ... | IV-36 |
| 4.2.1.1. | <i>Usecase Diagram</i> | IV-36 |
| 4.2.1.2. | <i>Usecase Diagram Spesification</i> | IV-37 |
| 4.2.1.3. | <i>Sequence Diagram</i> | IV-43 |
| 4.2.1.4. | <i>Class Diagram</i> | IV-44 |
| 4.2.1.5. | <i>Activity Diagram</i> | IV-46 |
| 4.3. | Perancangan Menu | IV-48 |
| 4.3.1. | Perancangan Struktur Menu Aplikasi | IV-48 |
| 4.3.2. | Perancangan Antarmuka Perangkat Berbasis Web ... | IV-49 |
| 4.4. | Perancangan <i>Database</i> | IV-53 |
| BAB V | IMPLEMENTASI DAN PENGUJIAN..... | V-1 |
| 5.1. | Implementasi | V-1 |
| 5.1.1. | Performansi Aplikasi..... | V-1 |
| 5.1.2. | Arsitektur Aplikasi | V-2 |
| 5.1.3. | Implementasi <i>Form Customer Service</i> Dispenda..... | V-3 |
| 5.1.4. | Implementasi <i>Form Utama Jasaraharja</i> | V-4 |
| 5.2. | Pengujian | V-6 |
| 5.2.1. | Pengujian Black Box..... | V-6 |
| BAB VI | PENUTUP | VI-1 |
| 6.1. | Kesimpulan..... | VI-1 |
| 6.2. | Saran | VI-1 |
| DAFTAR PUSTAKA | | xxii |
| LAMPIRAN | | xxii |
| RIWAYAT HIDUP | | xxiv |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini banyak perusahaan atau instansi yang menghadapi tantangan dalam memenuhi permintaan *customer* dengan layanan yang tinggi. Dalam lingkungan seperti ini bagaimana mengkombinasikan antar perkembangan teknologi serta globalisasi ekonomi pada perusahaan atau instansi pemerintahan. Banyak perusahaan yang menjalin kerjasama dengan menggabungkan sumber daya yang dimiliki suatu perusahaan dalam membuka kesempatan untuk mengelola sumber daya secara bersama sehingga dapat meminimalisir biaya, serta merespon permintaan dari masing-masing sumberdaya.

Adapun salah satu perusahaan atau instansi yang menghadapi tantangan dalam mengelola data informasi dengan layanan yang tinggi adalah PT. Jasa Raharja. Salah satu contoh kesulitan di PT. Jasa Raharja adalah pada Sub Bagian Pelayanan Klaim yang mengalami kesulitan dalam mengirim data ke Samsat dengan menggunakan *email* atau *fax*. Terdapat kesalahan dalam pengiriman data yang disebabkan oleh batasan dalam mengirim data via email atau fax.

Kebutuhan untuk membuat konsep kerjasama yang dinamis antar perusahaan seperti contoh tersebut, difokuskan pada sistem *enterprise* yang mengintegrasikan proses bisnis antar *enterprise*. Arsitektur sistem *enterprise* dikenal dengan istilah Service Oriented Architecture. SOA menyediakan suatu infrastruktur teknologi informasi yang fleksibel, efisien, serta efektif. Teknologi tersebut dapat meningkatkan performansi proses bisnis khususnya dalam mengelola data informasi.

SOA menurut Open Group adalah sebuah model membangun *web service* yang paling efektif. Tujuan utama dari SOA adalah untuk menunjang perusahaan dengan menggunakan teknologi informasi (TI) dengan cara membuat hubungan antar website yang efektif. SOA adalah sebuah jembatan yang menghubungkan antar perusahaan agar dapat sinergis. SOA merupakan hasil yang dapat dicapai dengan keselarasan yang lebih baik antara bisnis dan teknologi informasi (Arief, 2011).

Terdapat banyak teknologi yang dapat digunakan untuk mengimplementasikan arsitektur SOA, diantaranya adalah: CORBA, DCOM, RMI, dan *Web Service*. *Web service* saat ini sudah berkembang sangat pesat, *web service* membantu banyak perusahaan atau instansi pemerintahan dalam mengelola informasi perusahaan atau instansi yang dihubungkan pada perusahaan atau instansi lainnya secara terbuka menggunakan aplikasi berbasis *web*. *Web service* yang banyak diterapkan saat ini adalah dalam aplikasi *customer service*, dimana membantu perusahaan dalam bidang pelayanan terhadap suatu informasi yang ada di perusahaan atau instansi pemerintahan. *Web Service* dapat menyediakan layanan dengan baik, karena mampu menyediakan standar dari perbedaan aplikasi dan dapat berjalan di banyak platform, sehingga dapat digunakan untuk *integrasi system*, seperti: *Bisnis to Bisnis* (B2B), *Bisnis to Customer* (B2C), serta dalam aplikasi *e-business to Virtual Enterprise* (VE) (Arief, 2011).

Implementasi SOA berbasis *web service* seperti ini yang sangat dibutuhkan Sub Bagian Pelayanan Klaim di PT. Jasa Raharja dalam mengelola atau mengatasi informasi pelayanan tentang asuransi jiwa bagi masyarakat yang mana data informasi ini saling berkaitan dengan data yang ada di kepolisian setempat dan juga data di pemerintahan. Dengan menggunakan konsep layanan *web service* yang baru berkembang saat ini, selanjutnya dapat mengelola sumber daya dari berbagai informasi yang ada baik dari perusahaan asuransi, kepolisian atau pemerintahan secara efektif dan efisien.

1.2 Rumusan Masalah

Berdasarkan dari permasalahan latar belakang yang telah dijelaskan, maka rumusan masalah penelitian ini adalah menganalisa dan mengintegrasikan data pada dua instansi terkait terhadap pelayanan menggunakan metode SOA dengan teknologi *web service* pada aplikasi *customer service*.

1.3 Batasan Masalah

1. *Web service* yang dibuat khusus untuk pengelolaan data pada *customer service* di Dispenda.
2. Teknologi *web service* yang dibuat dapat diakses pada jaringan komunikasi internet atau jaringan komputer lokal menggunakan sistem pertukaran informasi berbasis XML.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini untuk mengintegrasikan data pada dua instansi terkait dalam mengelola data secara tepat dan cepat, serta dapat memberikan laporan secara berjangka.

1.5 Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan Tugas Akhir yang akan dibuat :

BAB I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan dari Tugas Akhir yang dibuat.

BAB II Landasan Teori

Bab ini membahas tentang teori-teori umum dan khusus yang berhubungan dengan tugas akhir ini.

BAB III Metodologi Penelitian

Bab ini membahas langkah-langkah yang dilaksanakan dalam proses penelitian, yaitu tahapan penelitian pendahuluan, identifikasi masalah, jadwal penelitian, studi literatur, wawancara, implementasi dan pengujian aplikasi, serta penarikan kesimpulan.

BAB IV Analisa dan Perancangan

Bab ini berisi pembahasan mengenai kebutuhan sistem, yang terdiri dari : Hasil Wawancara, Bizagi, Kandidat *Service*, Seleksi Kandidat *Service*, Arsitektur Sistem Lanjut, Identifikasi Arsitektur Sistem, *Usecase Diagram*, *Usecase Diagram Spesification*, *Sequence Diagram*, *Class Diagram*, *Activity Diagram*, *Deployment Diagram*, Perancangan Struktur Menu Aplikasi, Perancangan Antarmuka Perangkat Berbasis Web, Perancangan *Database* menggunakan pendekatan berorientasi objek.

BAB V Implementasi dan Pengujian

Bab ini berisi penjelasan mengenai implementasi yang terdiri dari: batasan implementasi, lingkungan implementasi, hasil implementasi, pengujian sistem dan kesimpulan pengujian.

BAB VI Kesimpulan dan Saran

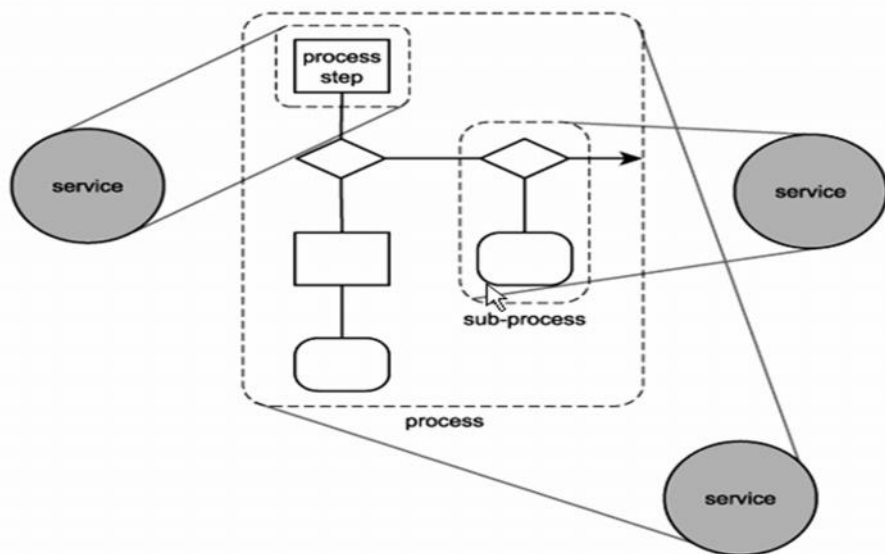
Bagian ini berisi kesimpulan yang dihasilkan dari pembahasan tentang analisa dan implementasi metode SOA menggunakan teknologi *web service* pada aplikasi *customer service*.

BAB II

LANDASAN TEORI

2.1 *Service Oriented*

Service oriented merupakan sebuah pendekatan dalam penyelesaian masalah besar dengan membaginya menjadi sekumpulan layanan (*service*) kecil yang menyelesaikan permasalahan secara keseluruhan (*spesifik*). Seperti dapat dilihat pada Gambar 2.1, lingkup dari *service* tidak terbatas, *service* dapat menyelesaikan melalui sebuah proses besar atau hanya satu langkah proses kecil. Hal ini dapat disesuaikan tergantung kebutuhan *service* (Erl, 2005)



Gambar 2.1. Enkapsulasi *Business Process* dengan *Service* (Erl, 2005)

Pembagian berdasarkan *service* ini sesungguhnya bukan sesuatu yang baru, karena telah banyak diterapkan.

2.2 Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) merupakan sebuah kumpulan *service*, *Service* dalam lingkup SOA adalah sekumpulan fungsi, prosedur, atau proses yang akan memberi respon yang diminta oleh sebuah *client* kepada *server*. *Service* merupakan satuan kerja yang dilakukan oleh penyedia *service* untuk mencapai hasil akhir yang diinginkan kepada *customer service*.

Definisi *Service Oriented Architecture* (SOA) menurut Open Group adalah sebuah *model architecture* yang mendukung *service orientation*. Definisi tersebut terfokus pada *model arsitektur*, *service orientation*, *service* serta fitur-fitur yang merupakan lingkup pada SOA. Organization for Advancement of Structured Information Standards (OASIS) mendefinisikan SOA sebagai paradigma yang digunakan untuk mengatur dan memanfaatkan kemampuan terdistribusi yang mungkin berada di bawah kendali kepemilikan suatu domain yang berbeda.

SOA didefinisikan oleh *World Wide Web Consortium* (W3C) sebagai suatu bentuk arsitektur sistem terdistribusi yang pada umumnya ditandai dengan *logical view*, *message orientation*, *description orientation*, *granularity*, dan *platform neutrality*. Pada tahun 2007 XML.com mendefinisikan SOA sebagai sebuah gaya arsitektur yang memiliki tujuan untuk mencapai *loosely couple* antar agen perangkat lunak yang berinteraksi. Empat karakteristik yang dimiliki oleh SOA berdasarkan Raghu Kodali antara lain :

- a. Antarmuka yang disusun dengan XML yang menggunakan WSDL.
- b. Skema XML yang disebut dengan XSD yang harus digunakan untuk mengolah pesan.
- c. *Registry* UDDI berdasarkan pada penyimpanan daftar *service* yang disediakan.
- d. Setiap *service* harus mempertahankan tingkat kualitas yang ditetapkan melalui persyaratan keamanan QoS.

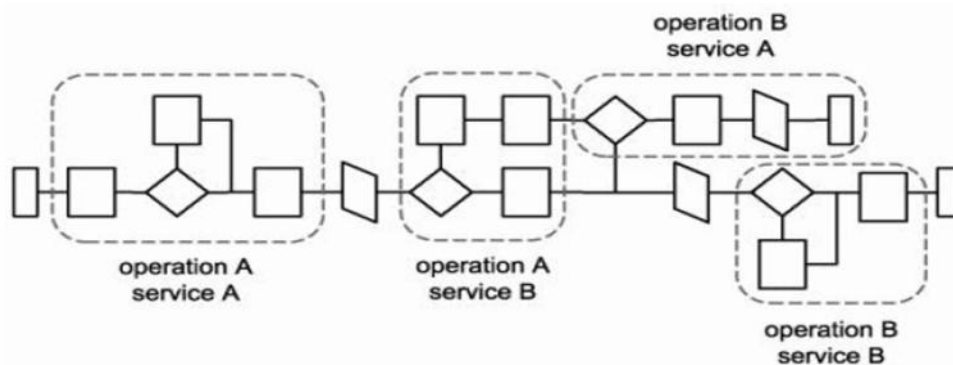
IBM mengusulkan bahwa SOA menggambarkan gaya arsitektur yang memperlakukan komponen perangkat lunak sebagai *service set*. Definisi tersebut ditegaskan sebagai kebutuhan bisnis yang harus mengendalikan definisi dari *service* dan nilai tujuan harus terfokus dengan fleksibilitas *service*.

2.3 Konsep *Service Oriented Architecture* (SOA)

Bila dilihat pada penjelasan sebelumnya, Erl (2005) memaparkan SOA terdiri atas sekumpulan *service*, namun sekumpulan *service* tidak cukup untuk membentuk sebuah *arsitektur*. SOA terdiri atas empat komponen, yaitu:

- a. *Message*, yaitu data yang dibutuhkan untuk menyelesaikan sebagian atau sebuah unit kerja, yang dipertukarkan antara satu *service* dengan *service* yang lainnya.
- b. *Operation*, yaitu fungsi-fungsi yang dimiliki oleh sebuah *service* untuk memproses *message* hingga menghasilkan sesuatu. Fungsi-fungsi inilah yang nantinya akan saling berinteraksi untuk menyelesaikan sebuah unit kerja.
- c. *Service*, mempersentasikan sekumpulan *operation* yang berhubungan untuk menyelesaikan sekumpulan unit kerja yang berhubungan.
- d. *Process*, merupakan *business rule* yang menentukan operasi mana yang digunakan untuk mencapai tujuan tertentu.

Ilustrasi dari keempat komponen ini dapat dilihat pada Gambar 2.2 (Erl, 2005).

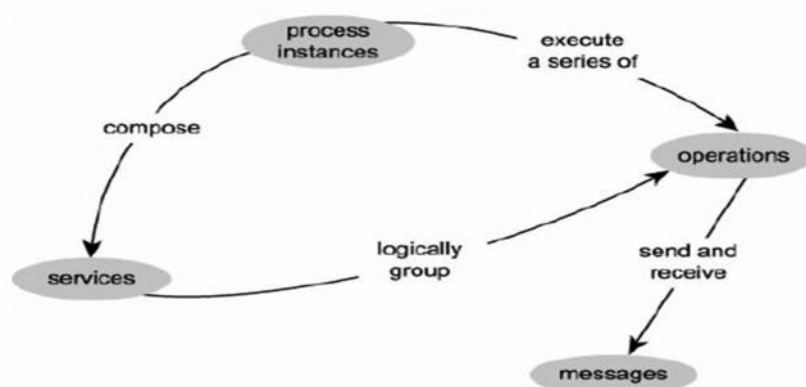


Gambar 2.2. Ilustrasi Operations dan Services (Erl, 2005)

Komponen-komponen tersebut terhubung satu sama lain dengan deskripsi sebagai berikut:

1. Sebuah *operation* mengirim dan menerima *message* untuk mengerjakan sesuatu.
2. Sebuah *operation* dapat didefinisikan oleh *message* yang memprosesnya.
3. Sebuah *service* mengelompokkan sekumpulan *operation* yang berhubungan.
4. Sebuah *service* didefinisikan oleh *operation* yang membentuknya.
5. Sebuah instans dari *service* dapat mengkomposisi *service* lain.
6. Sebuah instans dari *process* tidak harus didefinisikan oleh *service* karena mungkin hanya membutuhkan sebagian dari fungsionalitas yang diberikan oleh *service*.
7. Sebuah instans dari *process* memicu sekumpulan *operation* berjalan untuk menyelesaikan proses otomatisasi.
8. Setiap instans dari *process* didefinisikan secara parsial *operation* yang digunakannya.

Ilustrasi dari keterhubungan seluruh komponen dapat dilihat pada Gambar 2.3 (Erl, 2005).



Gambar 2.3. Keterhubungan antar Komponen SOA (Erl, 2005)

2.4 Prinsip - Prinsip *Service Oriented Architecture* (SOA)

Erl (2005) memaparkan beberapa prinsip yang seringkali digunakan terkait dengan pendekatan SOA yaitu:

Prinsip 1: *Service* dapat digunakan kembali (*Reusability*)

Pengembangan sistem yang menggunakan pendekatan SOA, *service* dirancang secara khusus untuk mendukung penggunaan kembali sesuai dengan kebutuhannya. Berupa data yang dibagi menjadi sekumpulan *service* yang dapat memudahkan *reuse*.

Prinsip 2: *Service* merupakan *formal specification*

Service tidak membutuhkan suatu pembagian apapun didalam pengembangan yang menggunakan pendekatan SOA untuk dapat berinteraksi dengan *service* lainnya. *Service* membutuhkan sebuah kontrak yang formal yang dapat mendeskripsikan setiap *service* yang telah ada dan menentukan persyaratan yang dibutuhkan pada pertukaran informasi yang terjadi.

Prinsip 3: *Service* merupakan *loosely couple*

Service secara khusus pada pendekatan SOA dirancang untuk dapat berkomunikasi dengan *service* lainnya tanpa perlu saling ketergantungan. Setiap *service* berdiri sendiri secara independen dan tidak tergantung *service* lain untuk berjalan. Ketergantungan diminimalisir sehingga hanya butuh mekanisme komunikasi satu sama lain.

Prinsip 4: Inti sari *service* berdasarkan logika

Satu-satunya bagian dari *service* yang terlihat di dunia luar pada penerapan pendekatan SOA merupakan hal-hal yang ditampilkan melalui kontrak *service* tersebut. Logika dasar yang melampaui hal tersebut dinyatakan ke dalam deskripsi yang terdiri dari kontrak yang tidak nyata

dan tidak relevan dengan permintaan dari service tersebut. Setiap service memiliki kesepakatan mengenai cara untuk komunikasi.

Prinsip 5: *Decomposition Service* bersifat *Abstraction*

Penggunaan SOA menyebabkan *service* dapat menyusun *service* yang lain. Hal ini memungkinkan logika yang dapat digambarkan pada tingkat *granularity* yang berbeda dan mempromosikan penggunaannya kembali serta penyusunan dari inti sari yang berada pada *layer.Service* tidak memperlihatkan bagaimana logik diimplementasi di dalamnya.

Prinsip 6: *Service* bersifat *Autonomy*

Logika yang menggunakan pendekatan SOA dipengaruhi oleh sebuah *service* yang diletakan pada sebuah batasan yang tidak dapat dilihat. *Service* tersebut akan mengontrol batas tersebut dan untuk mengeksekusinya tidak perlu bergantung dengan *service* lain. *Service* memiliki hak penuh terhadap semua data yang dienkapsulasi.

Prinsip 7: *Service* bersifat *Statelessness*

Service yang berbasiskan SOA tidak harus membutuhkan pengaturan informasi *state*. Hal ini dikarenakan *state* tersebut dapat menghalangi kemampuan *service* untuk bergabung atau berintegrasi. *Service* tidak memilki status tertentu terkait dengan aktivitas yang dilakukannya.

Prinsip 8: *Service* tidak terdeteksi bersifat *Discoverability*

Service yang dirancang harus dapat memungkinkan deskripsi mengenai diri *service* sendiri di dalam sistem yang telah menerapkan SOA untuk dapat menemukan *service* tersebut dan dapat dimengerti oleh manusia dan pemohon *service* tersebut yang dapat menggunakan logika dalam *service.Service* didesain untuk deskriptif sehingga bisa ditemukan dan diakses melalui mekanisme pencarian tertentu.

Dari perspektif teknologi informasi, fungsi-fungsi yang terdapat pada sebuah enterprise terbagi menjadi 2 bagian yakni *business logic* dan *application logic*. *Business logic* adalah sebuah implementasi dari *business requirements* atau kebutuhan bisnis yang terdokumentasi dari area bisnis sebuah enterprise. Logika bisnis distrukturkan menjadi beberapa proses yang merepresentasikan requirements dari bisnis, keterbatasan, ketergantungan antar proses dan pengaruh dari luar enterprise. Sedangkan, logika aplikasi adalah suatu bentuk implementasi yang terotomasi dari logika bisnis yang terorganisasi menjadi berbagai solusi teknologi. Logika aplikasi mengekspresikan aliran kerja dari proses bisnis melalui suatu sistem yang dikembangkan melalui infrastruktur teknologi informasi dari sebuah enterprise.

Service oriented architecture memperkenalkan satu lapisan baru yang disebut lapisan service, yang berfungsi untuk mengenkapsulasi logik dari aplikasi terhadap logik dari bisnis. Hal ini terkait dengan salah satu karakteristik SOA, yakni meningkatkan aspek *loosely coupled* antara logik bisnis dan logik aplikasi. Ketika lapisan service dapat merepresentasikan logik dari bisnis dan aplikasi, maka ketergantungan langsung antara logik bisnis dan logik aplikasi rendah. Lapisan antarmuka service (*service interface layer*) merupakan suatu abstraksi dari logik bisnis, aplikasi, dan aliran proses secara keseluruhan. *Service interface layer* memiliki 3 layer abstraksi yaitu :

1) *Application Service Layer*

Layer ini berperan dalam membangun dasar dari *service interface layer* yang berfungsi untuk merepresentasikan fungsionalitas yang terkait dengan teknologi secara spesifik. *Service* yang berada pada layer ini disebut dengan *application service*.

2) *Business Service Layer*

Business service layer adalah suatu layer *service* yang merepresentasikan *business logic*. *Business service* yang terdapat pada

layer ini berperan dalam mengekspresikan logik bisnis melalui pendekatan *service oriented*.

3) Orchestration Service Layer

Orchestration service layer merupakan sebuah layer yang menyediakan abstraksi dengan level tertinggi yang menjawab kebutuhan dari *service* lain untuk memastikan bahwa *operasi service* dieksekusi dalam urutan yang ditentukan. Pada layer ini semua proses bisnis yang ada di dalam sistem didefinisikan dan dijalankan dengan menggunakan fungsi yang terdapat pada *business service layer*.

2.5 Web Service

Web Service adalah sistem software yang dirancang untuk mendukung sistem yang berinteraksi melalui jaringan. *Web Service* memiliki method (operasi) penyedia data yang dipertukarkan dengan menggunakan protokol HTTP dan bahasa XML (Hamids dalam Priyambodo, 2005). *Web Service* merupakan sebuah sistem, implementasi *Web Service* dapat dilakukan pada jaringan komunikasi internet atau jaringan komputer lokal. *Web Service* merupakan perangkat lunak yang didesain untuk mendukung interaksi operasi mesin-mesin melalui jaringan. Memiliki antarmuka yang dijelaskan dalam format mesin yang dapat diproses *Web Services Description Language* (WSDL). Sistem lain berinteraksi dengan *Web Service* dengan cara yang ditentukan oleh deskripsi menggunakan pesan SOAP, biasanya disampaikan menggunakan HTTP dengan menggunakan XML dalam hubungannya dengan standar Web yang terkait lainnya.

Web Service biasanya menggunakan *Extensible Markup Language* (XML) pesan yang mengikuti standar SOAP. *Web services* adalah sistem pertukaran informasi berbasis XML yang menggunakan internet untuk interaksi antar aplikasi. Teknologi ini merupakan standar yang diadopsi oleh banyak vendor perangkat lunak, karena memiliki standar terbuka. Standar terbuka ini yang memungkinkan aplikasi *web service* yang diimplementasi oleh vendor berbeda dapat berkomunikasi satu sama lain. Perkembangan *web services* yang begitu

cepat menyebabkan lahirnya ekstensi-ekstensi *web services* yang memperluas fungsi *web services* itu sendiri.

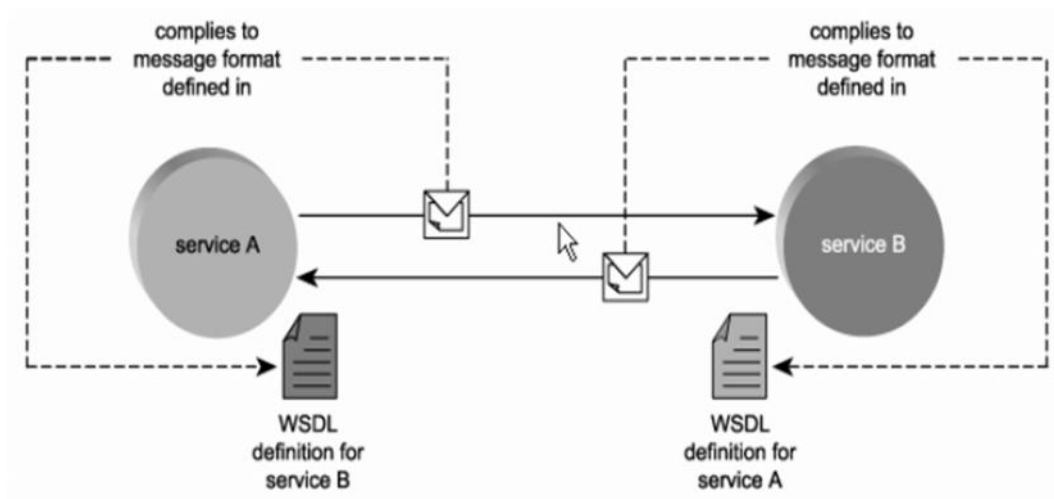
2.6 Service Oriented Architecture (SOA) dan Web Service

Seperti telah dijelaskan sebelumnya, untuk mengimplementasikan sebuah *service*, terdapat beberapa keutamaan, yaitu bagaimana *service* berkomunikasi, bagaimana *service* didesain, dan bagaimana pesan antar *service* didefinisikan. *Web Service* merupakan sebuah teknologi yang dapat memenuhi kebutuhan setiap *service* (Erl, 2005). *Web Services* sebagai sebuah teknologi dapat memfasilitasi beberapa perancangan seperti berikut :

- a. Sebuah *service* dalam SOA adalah sebuah aplikasi *Web Service*.

Pada dasarnya sebuah *service* didalam SOA adalah sebuah aplikasi. Aplikasi ini mempresentasikan sebuah *business logic* atau *automation logic* dari sebuah proses sistem besar yang mencakupinya. Hal itu sudah dimiliki oleh sebuah aplikasi *Web Services*. Maka dari itu, implementasi *Service* dalam SOA merupakan aplikasi *Web Service*.

- b. Hubungan satu *service* dengan yang lainnya didefinisikan dengan *Web Service Description Language* (WSDL).



Gambar 2.4. Peran WSDL dalam Hubungan antar *Service* (Erl, 2005)

WSDL mendeskripsikan bagaimana format sebuah pesan yang akan dikirim oleh sebuah *Web Service* yang harus ditulis, agar dapat dimengerti oleh *Web Service* lainnya. Dalam sudut pandang SOA, yang menggunakan *Web Service* sebagai *Service*, teknologi WSDL ini menjadi jembatan untuk menghubungkan sebuah *Service* ke *Service* lainnya. Ilustrasi hubungan ini dapat dilihat pada Gambar 2.4 (Erl, 2005).

c. *Service* berkomunikasi satu sama lain menggunakan SOAP *messaging*.

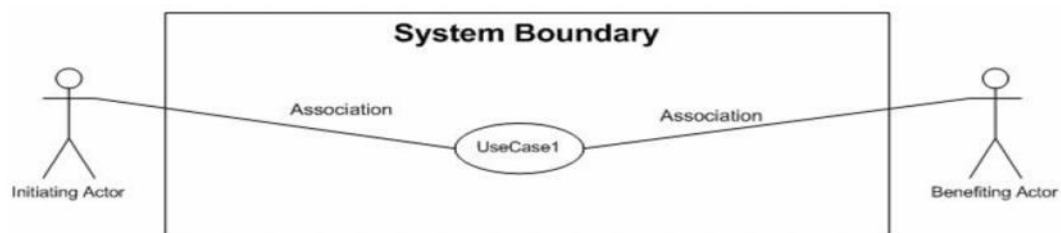
Kebutuhan untuk saling berkirim pesan dalam SOA merupakan hal yang paling penting. Untuk teknologi pengiriman pesan, kita dapat menggunakan SOAP. Hal ini dikarenakan SOAP *fleksibel* dan dapat diperluas.

2.7 Unified Modelling Language (UML)

UML adalah sebuah bahasa untuk menentukan, visualisasi konstruksi, mendokumentasikan *artifacts* (model, deskripsi, atau *software*) dari sistem *software*, untuk memodelkan bisnis, dan sistem *non-software* lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks.

a. Usecase Diagram

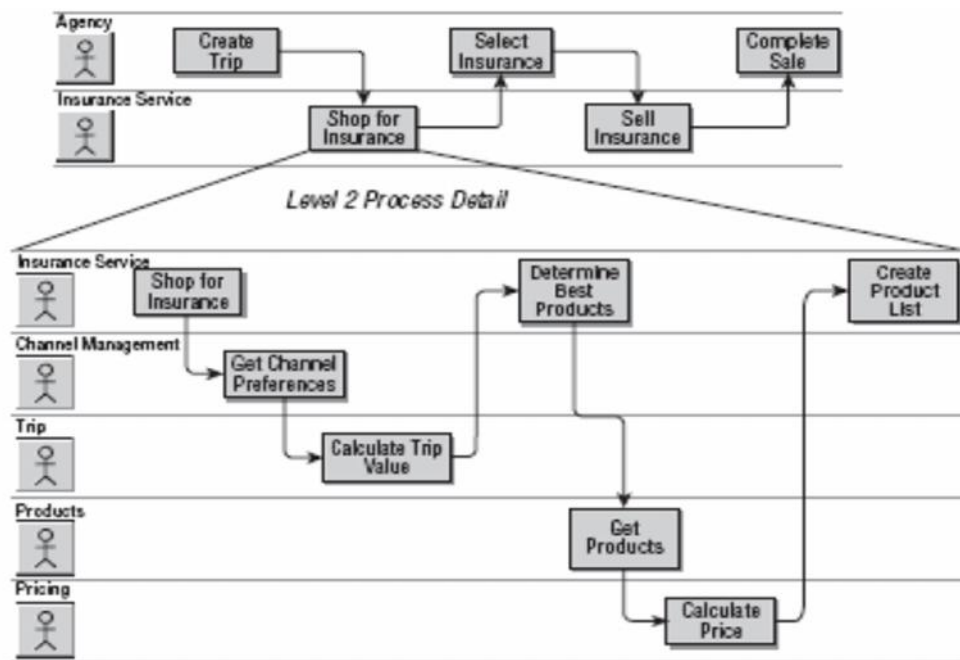
Menurut Whitten et al, *Usecase Diagram* adalah diagram yang menggambarkan interaksi antara sistem, eksternal sistem, dan user. Dengan kata lain, diagram ini menjelaskan siapa yang akan menggunakan sistem tersebut, dan bagaimana cara user tersebut berinteraksi dengan sistem.



Gambar 2.5. Usecase Diagram (Joseph Schmuller, 1999)

b. Process Diagram

Process Diagram digunakan untuk menggambarkan suatu tingkat lebih detail dari proses bisnis utama dimana user dan proses dibuat detail untuk satu alur proses pekerjaan dan kemungkinan yang dapat terjadi. Skenario yang terjadi tersebut yang menggambarkan satuan proses bisnis dari suatu bagian kegiatan. Contoh *process diagram* yaitu sebagai berikut :



Gambar 2.6. *Process Diagram* (R. Mike, 2008)

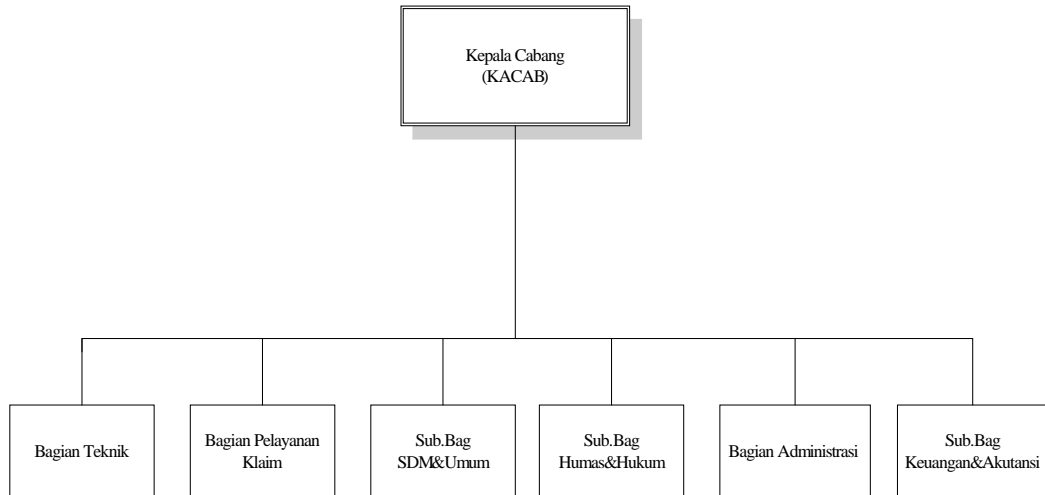
2.8 Customer Service

Customer service adalah suatu rangkaian kegiatan yang dirancang untuk meningkatkan nilai kepuasan pelanggan yang meliputi tingkat kepercayaan dan *informatif*. Adapun dari sudut pandang bisnis, *Customer Service* memberikan pengaruh yang cukup penting dalam meningkatkan pendapatan dari suatu perusahaan atas barang atau jasa yang dijual kepada konsumen.

2.9 PT. Jasa Raharja (Persero)

a. Struktur Organisasi

Struktur Organisasi Internal :



Gambar 2.7. Tempat Pelaksanaan

b. Dasar Hukum

Penunjukan Perusahaan Negara oleh Menteri berdasarkan dengan Surat Keputusan Menteri Urusan Pendapatan, Pembiayaan dan Pengawasan Republik Indonesia No. BAPNI-3-3, yang menunjuk Perusahaan Negara Asuransi Kerugian Jasa Raharja untuk melaksanakan penyelenggaraan Dana Pertanggungan Wajib Kecelakaan Penumpang dan Dana Kecelakaan Lalu Lintas Jalan sesuai dengan ketentuan ketentuan yang telah diatur dengan Undang-undang No. 33 tahun 1964, dan Undang-undang No. 34 tahun 1964 dan Peraturan Pemerintah No. 8 tahun 1965. Surat Keputusan Menteri tersebut ditetapkan pada tanggal 30 Maret 1965 namun demikian mulai berlaku dengan daya surut pada tanggal 1 Januari 1965.

Perusahaan negara Jasa Raharja itu sendiri telah didirikan berdasarkan Undang undang No. 19 Prp. 1960 dengan suatu Peraturan Pemerintah No. 8 Tahun 1965 (Lembaran Negara Tahun 1965 nomor 14) yang juga mulai berlaku pada tanggal 1 Januari 1965.

Sejak tahun 1969 dengan dibentuknya Undang-undang No. 9 tahun 1969 tentang : bentuk-bentuk Usaha Negara, semua usahausaha Negara yang berbentuk Perusahaan dibedakan menjadi tiga jenis yaitu :

- 1) Perusahaan Jawatan disingkat (Perjan).
- 2) Perusahaan Umum disingkat (Perum).
- 3) Perusahaan Perseroan disingkat (Persero).

c. Fungsi PT. Jasa Raharja (Persero)

Fungsi PT. Jasa Raharja (Persero) ini berorientasi pada perintah Undang-undang No. 33 Tahun 1964 dan undang-undangNo. 34 tahun 1964 tentang iuran dan sumbangan wajib untuk dipupuk dan di himpun dan selanjutnya disalurkan kembali kepada masyarakat yang mengalami kecelakaan, sebagai asuransi jasaraharja. Asuransi jasa raharja adalah perlindungan dan jaminannegara kepada rakyatnya yang mengalami kecelakaan, sedang objeknya adalah manusia dan asuransi ini memberikan jaminanterhadap kerugian yang disebabkan oleh kecelakaan. Kerugian yang timbul dari kecelakaan dapat berupa meninggal, cacat sementara, cacat tetap, biaya pengobatan dan perawatan di rumah sakit.

Fungsi utama PT. Jasa Raharja adalah dengan pemupukan dana-dana yang cara pemupukannya dilakukan dengan mengadakan Iuran dan Sumbangan wajib tersebut adalah hanya golongan atau mereka yang berada atau mampu saja, sedang hasil pemupukannya akan dilimpahkan juga kepada perlindungan jaminan rakyat banyak. Oleh karena itu jaminan sosial rakyatlah yang dalam pada itu menjadi pokok tujuan yang utama.

d. Kegiatan PT. Jasa Raharja

Kegiatan PT Jasa Raharja (Persero) Pertanggungan wajib adalah karena ada salah satu pihakyang mewajibkan kepada pihak lain dalam mengadakan pertanggungan itu. Pihak yang mewajibkan ini biasanya ialah pihak pemerintah, tetapi tidak selalu dimonopoli pemerintah. Pihak pemerintah dalam hubungan hukum pertanggungan ini adalah sebagai penanggung. Pemerintah dalam mengambil tindakan mewajibkan itu

biasanya didasarkan atas pertimbangan melindungi golongan-golongan lemah, dari bahaya-bahaya yang menimpanya atau memberikan jaminan sosial atau *social security* bagi masyarakat.

Oleh sebab itu PT Jasa Raharja (Persero) dalam kegiatannya adalah memupuk dana Iuran Wajib dan Sumbangan Wajib dari masyarakat untuk selanjutnya disalurkan melalui santunan kecelakaan lalu lintas. Undang-undang No. 33 Tahun 1965 Pasal 3 ayat (1) a. Tiap penumpang yang sah dari kendaraan bermotor umum, kereta api, pesawat terbang, perusahaan penerbangan nasional dan kapal perusahaan perkapalan/pelayaran nasional, wajib membayar iuran melalui perusahaan yang bersangkutan untuk menutup akibat keuangan disebabkan kecelakaan penumpang dalam dalam perjalanan.

Undang-undang No. 34 Tahun 1964 Pasal 2 ayat (1) Pengusaha Pemilik alat angkutan lalu lintas jalan diharuskan memberi Sumbangan tiap tahun kepada Dana sebagaimana dimaksud dalam Pasal 1 (Dana ialah dana yang terhimpun dari sumbangan wajib yang dipungut dari para pemilik/pengusaha alat angkutan lalu lintas jalan dan yang disediakan untuk menutup akibat keuangan karena kecelakaan lalu lintas jalan korban/ahli waris yang bersangkutan). Pasal 4 ayat (1) setiap orang yang menjadi korban mati atau cacat tetap akibat kecelakaan yang disebabkan oleh alat angkutan lalu lintas jalan tersebut dalam Pasal 1, akan memberi kerugian kepadanya atau kepada ahli warisnya sebesar jumlah yang ditentukan berdasarkan Peraturan Pemerintah.

Selanjutnya PT Jasa Raharja (Persero) untuk dapat mengatur penggunaan dana terhimpun tersebut secara efektif dan efisien, perlulah dana-dana yang dapat diinvestasikan itu, dipusatkan dalam suatu badan Pemerintah. Suatu Perusahaan Negara yaitu PT Jasa Raharja (Persero) yang harus mengadministrir dana-dana tersebut secara baik sehingga terjaminlah kedua tujuan dari pemupukan dana-dana tersebut.

2.10 Tugas Pokok PT. Jasa Raharja

Tugas Pokok Perusahaan Jasa Raharja (Persero) adalah menghimpun dan memupuk dana masyarakat melalui iuran dan sumbangan wajib, untuk selanjutnya menyalurkannya kembali kepada masyarakat yang berwujud santunan jasa raharja, terhadap korban kecelakaan lalu lintas.

a. Sub Bagian Pelayanan Klaim :

Sub Bagian Pelayanan Klaim mempunyai tugas dan wewenang yaitu sebagai berikut:

- a. Bagian yang berhubungan langsung dengan pihak internal maupun *external* yang berhubungan dengan pelayanan di PT. Jasa Raharja (Persero).
- b. Kelancaran dan ketertiban pelaksanaan pekerjaan serta pengamanan sumber daya informasi di dalam dan di luar lingkungan PT. Jasa Raharja (Persero).
- c. Kelancaran pelayanan penunjang kegiatan operasional di kantor cabang.
- d. Terselenggaranya kegiatan bidang pelayanan dan pola kearsipan data di kantor cabang.

2.11 Kerjasama PT. Jasa Raharja

a. Kerjasama dengan Pemerintah Daerah Kabupaten dan Kota

Undang-undang No. 34 Tahun 1964 dan PP No. 18 Tahun 1965, telah mengamanatkan kepada PT. Jasa Raharja (Persero) untuk bekerjasama dengan Pemerintah Daerah dalam kaitannya dengan pemupukan dana sumbangan wajib. Kerjasama mencapai pada tingkat para petugas yang berada di setiap SAMSAT Kabupaten/Kota. Dilain pihak pelayanan pada Kantor Bersama ini juga sangat membantu masyarakat dalam upaya sadar hukum dalam memenuhi kewajibannya sebagai warga Negara yang taat akan hukum, khususnya dalam hal pembayaran pajak kendaraan bermotor dan kewajiban membayar sumbangan wajib jasa raharja. Selanjutnya masyarakat dalam memenuhi kewajibannya merasa mudah dan cepat

dengan adanya hubungan kerja sama yang saling mendukung antara PT. Jasa Raharja (Persero) dengan Pemerintah Daerah.

b. Kerjasama dengan Polri

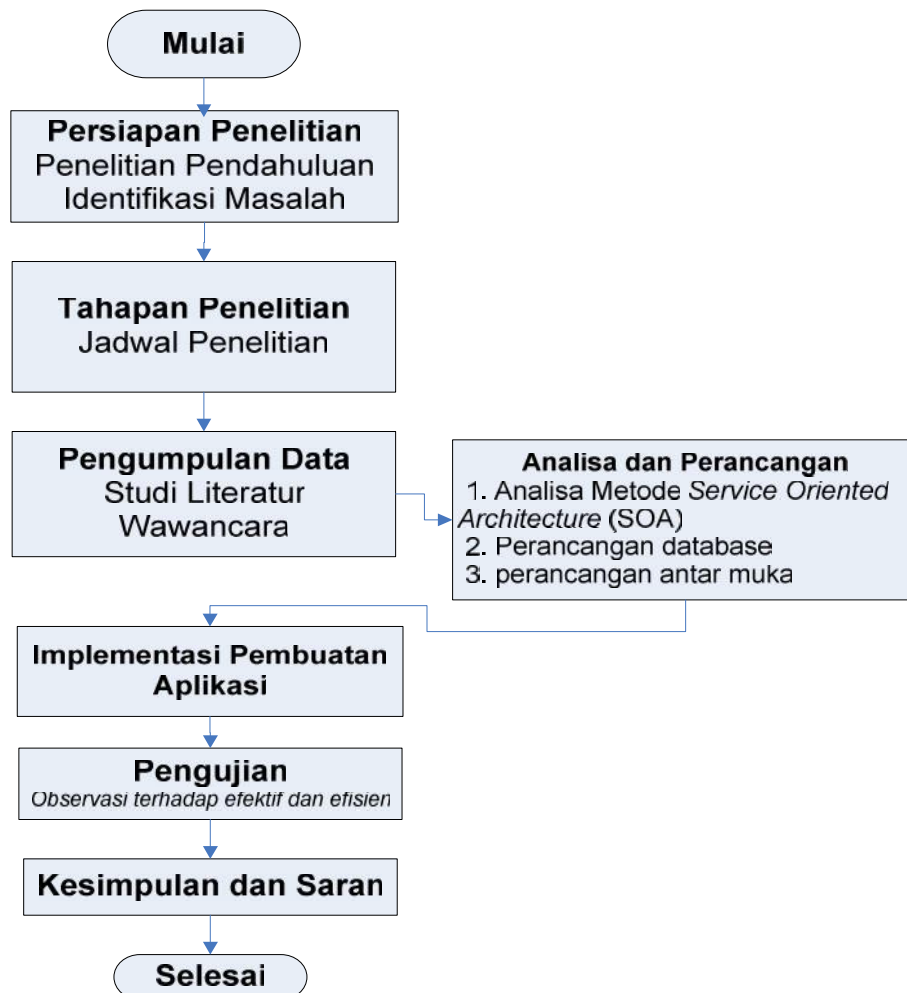
Dalam upaya melaksanakan tugas dan tanggung jawab di bidang operasional dan pelayanan berdasarkan Undang-Undang No. 33 dan 34 Tahun 1964 maka PT. Jasa Raharja (Persero) menjalin hubungan dengan Kepolisian. Pelaksanaan kerjasama antara Polri dengan PT Jasa Raharja, sudah selayaknya untuk lebih ditingkatkan karena fungsi dan tugas Polri khususnya Polantas sangat erat kaitannya dengan tugas dan fungsi PT. Jasa Raharja (Persero). Karena tugas dan tanggung jawab PT. Jasa Raharja (Persero) diantaranya adalah menyalurkan santunan jasa raharja kepada korban/ahli waris korban kecelakaan di jalan raya.

Fungsi peranan Polisi Lalu Lintas sangat penting dalam memberikan dukungan administrasi kelengkapan dan persyaratan akan hak santunan jasa raharja yang dibutuhkan oleh masyarakat yang mengalami kecelakaan di jalan raya. Dengan demikian maka kerjasama antara Kepolisian Republik Indonesia (Khususnya Polisi Lalu Lintas) dengan PT. Jasa Raharja (Persero) dapat berjalan baik, terutama dalam memupuk iuran dan sumbangan wajib dari masyarakat hingga penyalurannya kepada masyarakat melalui santunan jasa raharja.

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan tentang tahapan-tahapan yang dilakukan pada proses penelitian agar berjalan sesuai dengan tujuan yang telah ditentukan sehingga mencapai hasil yang baik. Penelitian ini menggunakan metodologi *Design Science Research* (DSR) yang memfokuskan pada pengembangan dan peningkatan kinerja model/desain dari sebuah sistem (Peffer, 2007). Tahapan penelitian dapat dilihat dari Gambar 3.1 berikut :



Gambar 3.1.Tahapan Metodologi Penelitian

3.1 Persiapan Penelitian

a. Penelitian Pendahuluan

Tahapan penelitian pendahuluan merupakan tahapan pencarian informasi awal mengenai penelitian-penelitian yang sudah dilakukan oleh peneliti-peneliti terdahulu, dengan cara mencari referensi-referensi dari buku-buku dan literature yang berhubungan dengan pembuatan aplikasi *Service Oriented architecture* (SOA) dengan menggunakan metode *web service*.

b. Identifikasi Masalah

Tahapan identifikasi masalah dilakukan setelah mendapatkan informasi-informasi yang dibutuhkan dari penelitian pendahuluan. Tahapan ini dilakukan untuk mengetahui permasalahan yang terjadi. Dari penelitian pendahuluanya itu diperlukan sebuah *Web Service*.

Berdasarkan dari identifikasi permasalahan yang telah dijelaskan, maka penulis merumuskan penelitian ini akan membahas mengenai implementasi *service Oriented Architecture* (SOA) menggunakan teknologi *Web Service* pada aplikasi *Customer Service*.

3.2 Tahapan Penelitian

Jadwal penelitian pengerjaan Tugas Akhir ini dijelaskan pada tabel 3.2 di bawah ini :

| No | Kegiatan | Waktu pelaksanaan bulan | | | |
|----|---|-------------------------|------|------|------|
| | | Ke 1 | Ke 2 | Ke 3 | Ke 4 |
| 1 | Persiapan Penelitian dan Pengumpulan Data | | | | |
| 2 | Analisa dan Perancangan | | | | |
| 3 | Implementasi dan Pengujian Aplikasi | | | | |
| 4 | Kesimpulan dan Saran | | | | |

Gambar 3.2.Jadwal Kegiatan

3.3 Pengumpulan Data

Tahapan pengumpulan data merupakan tahapan yang digunakan untuk memperoleh informasi dan data dari permasalahan yang ada. Ada beberapa tahapan dalam pengumpulan data ini, yaitu :

a. Studi Literatur

Studi Literatur merupakan salah satu tahap yang dilakukan dengan cara mencari referensi-referensi yang berhubungan dengan pembuatan *Web Service*. Referensi dapat berupa buku, jurnal, ataupun *browsing* di internet mengenai penelitian ilmiah yang pernah dilakukan sebelumnya. Studi literatur yang dilakukan berupa tinjauan berbagai literatur yang berkaitan dengan permasalahan yang akan dibahas secara keseluruhan mengenai pembuatan *Web Service*. Selain itu, literature yang ditinjau juga berkaitan dengan metode *Service Oriented Architecture* (SOA).

b. Wawancara

Melakukan wawancara secara langsung kepada Sub Bag Pelayanan untuk mengetahui tentang sistem yang digunakan selama ini, selanjutnya membuat penyelesaian permasalahan yang dihadapi mengenai pembuatan website yang diinginkan. Wawancara dilakukan secara keseluruhan mengenai hal-hal yang berkaitan dengan pembuatan *Web Service*.

3.4 Analisa dan Perancangan Sistem

a. Analisa Metode *Service Oriented Architecture* (SOA)

Analisa metode dilakukan untuk menjelaskan langkah-langkah yang akan dikerjakan terutama Implementasi *Service Oriented Architecture* (SOA) dengan menggunakan teknologi *web service*.

b. Perancangan Database

Setelah menganalisa bagaimana sistem yang akan dibuat maka dilakukan analisa terhadap rancangan basis data untuk melengkapi komponen sistem. Lokasi basis data nantinya akan berada di perangkat Berbasis Web sendiri karena menggunakan *database MySQL*.

c. Perancangan Antar Muka (*Interface*)

Merancang *interface* sistem bertujuan memudahkan komunikasi antara sistem dengan pengguna. Perancangan *interface* ditekankan pada tampilan yang baik, mudah dipahami dan menggunakan *navigasi-navigasi* yang mudah dimengerti.

3.5 Implementasi dan Pengujian

Pada tahapan ini, penulis akan menguji sistem yang telah diimplementasikan apakah sesuai dengan rancangan yang telah ditetapkan, berikut hal penting dalam pengujian.

- a. Pengujian *Web Service*, menguji apakah *Web Service* yang dibangun sesuai dengan rancangan dan keluaran yang diharapkan.
- b. Pengujian Implementasi *Service Oriented Architecture* (SOA) menggunakan *Web Service* pada Aplikasi yang dibuat.

3.6 Penarikan Kesimpulan

Setelah sistem selesai dibangun, maka tahap selanjutnya adalah penarikan kesimpulan. Penarikan kesimpulan berisi penjelasan mengenai hubungan antara metode yang digunakan dan hasil dari *Web Service*, apakah sesuai dengan tujuan atau tidak. Pada tahap ini akan diberikan hasil akhir dalam pembuatan *Web Service* ini.

BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisa

Analisa perangkat lunak dibutuhkan sebagai bahan acuan dalam membuat perancangan aplikasi. Analisa yang dilakukan dalam penelitian ini adalah menganalisa mengenai tahapan pembuatan aplikasi untuk menguji tingkat akurasi dari implementasi *Service Oriented Architecture* (SOA) menggunakan teknologi *web service* untuk memperoleh pengertian yang tepat dan pemahaman terhadap keseluruhan masalah, data, dan proses yang terkait dalam penelitian. Analisa fokus kepada parameter-parameter apa saja yang menjadi *input*, serta bagaimana proses *output* dihasilkan.

4.1.1 Analisa Sistem Lama

Sub Bagian Teknik dalam mengelola data yang didapat dari Dispenda membutuhkan waktu yang lama dalam memperoleh data yang diminta, karena dalam memperoleh data masih dilakukan menggunakan sistem secara manual. Dengan demikian pengelolaan data dapat dikatakan tidak efektif dan memerlukan waktu yang lebih lama pada sistem kerja tersebut.

Analisa sistem lama biasanya mengharuskan Sub Bagian Teknik untuk menelepon, mengemail, dan mengunjungi Dispenda untuk dapat berkonsultasi mengenai Datako di Kantor Samsat. Datako merupakan data pembayaran pajak kendaraan bermotor yang dilakukan secara periodik (setiap tahun) di kantor Samsat pada saat pendaftaran atau perpanjangan STNK.

Berdasarkan hasil wawancara yang di lakukan dengan Customer Service (DISPENDA) dan Sub Bagian Teknik (PT. JASARAHARJA), dapat dirangkum proses bisnis kerja yang berlangsung di kedua instansi tersebut. Adapun proses bisnis nya dapat dilihat pada Tabel 4.1 berikut.

Tabel 4.1 Hasil Wawancara Dispenda

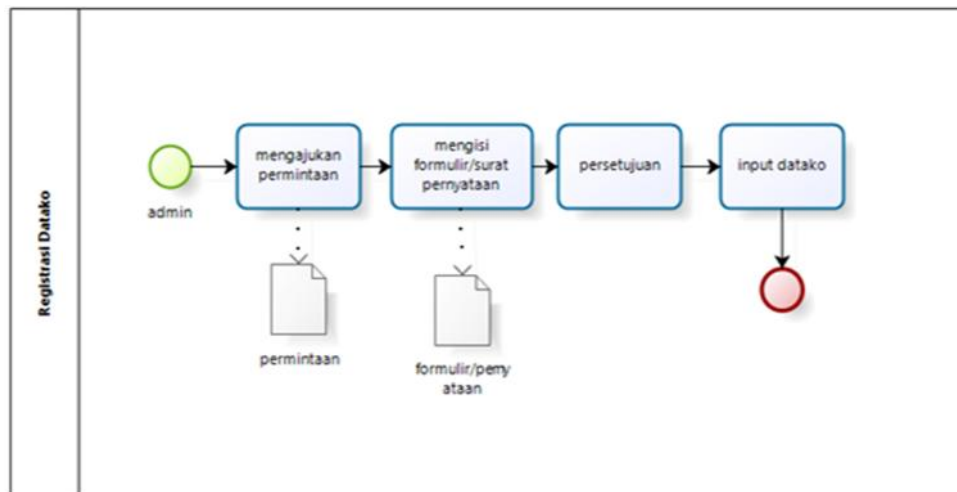
| NO | Sub. Bagian | Tugas Pokok | Proses Bisnis | Sub Proses |
|----|-------------------------|--|--|------------|
| | Customer Service | 1) Mengendalikan serta menyimpan arsip-arsip atau surat. | a) Pembayaran SW dilakukan secara periodik (setiap tahun) di kantor Samsat pada saat pendaftaran atau perpanjangan STNK. b) Setiap penumpang yang akan menggunakan alat transportasi umum membayarkan iuran wajib yang disatukan dengan ongkos angkut pada saat membeli karcis atau membayar tarif angkutan dan pengutipan ini dilakukan oleh masing-masing operator (pengelola) alat transportasi tersebut. c) Selanjutnya data pembayaran SW (sumbangan wajib) diserahkan pada pihak Jasa Raharja. | - |

Tabel 4.2 Hasil Wawancara PT. Jasaraharja

| NO | Sub. Bagian | Tugas Pokok | Proses Bisnis | Sub Proses |
|----|---------------------------------------|---|---|----------------------------|
| | Sub Bagian Pelayanan dan Klaim | 1) Bagian yang berhubungan langsung dengan pihak internal maupun external perusahaan. | a) Mengajukan permintaan data pada Sub. Bagian Teknik. b) Mendapatkan data (DATAKO) dari Sub. Bagian Teknik. | Hasil input data (DATAKO). |

Detail hasil wawancara proses bisnis yang sedang berjalan di Customer Service (DISPENDA), Sub Bagian Teknik, dan Sub Bagian Pelayanan (PT. JASARAHARJA) dapat dilihat pada lampiran C.

Dari hasil wawancara proses bisnis, dibuat alur kerja (*workflow*) yang di gambarkan untuk setiap bisnis-bisnis yang berjalan di Customer Service (DISPENDA), Sub Bagian Teknik, dan Sub Bagian Pelayanan (PT. JASARAHARJA). Adapun tahapan-tahapan yang sedang berjalan, dapat di gambarkan sebagai berikut :



Gambar 4.1 Registrasi Datako (DISPENDA)

Tabel 4.3 Proses Registrasi Datako (DISPENDA)

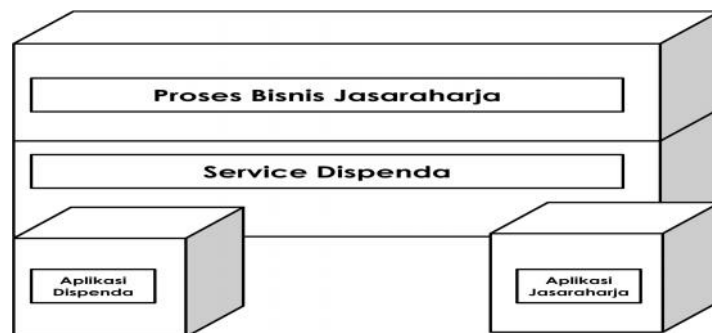
| Proses | Sub Proses | Kegiatan |
|-----------------------------------|--|---|
| Mengajukan permintaan | a) Mengajukan permintaan data ke Dispenda di Samsat | Sub. Bagian Teknik PT. Jasa Raharja (Persero) mengajukan permintaan kepada <i>Customer Service</i> Dispenda yang ada di Samsat. |
| Mengisi formulir/surat pernyataan | 1) Mengisi formulir 2) Mengisi surat pernyataan | Mengisi formulir/surat pernyataan yang ditujukan kepada <i>Customer Service</i> Dispenda di Samsat. |
| Persetujuan | 1) Penerimaan formulir/surat pernyataan 2) Persetujuan permintaan | <i>Customer Service</i> Dispenda di Samsat menerima formulir/surat pernyataan, serta memberikan persetujuan. |
| Input datako | 1) Memberikan data 2) Melakukan input datako | Sub. Bagian Teknik PT. Jasa Raharja (Persero) menerima data berupa Hard Copy, Selanjutnya menginputkan |

| | | |
|--|--|--------------------------|
| | | data tersebut ke sistem. |
|--|--|--------------------------|

Untuk detail alur kerja (*workflow*), tahapan-tahapan lainnya yang sedang berjalan di Dispenda, dan PT. Jasaraharja dapat dilihat pada lampiran C.

4.1.2 Analisa Sistem Baru

Terdapat dua bagian penting dalam kebutuhan sistem yang akan dibangun yaitu dari perangkat Berbasis Web sebagai platform yang akan digunakan, dan database yang digunakan untuk menyimpan dan memperbaharui data yang dapat dilakukan oleh pengguna sistem menggunakan teknologi *web service*.



Gambar 4.2 Implementasi *Layer* pada *Enterprise*

Fungsi yang terdapat pada Dispenda dan Jasaraharja berupa *business logic* dan *application logic*. Analisa proses bisnis dilakukan, untuk dapat mengintegrasikan data pada kedua perusahaan. Proses bisnis yang berlangsung pada aplikasi Dispenda dan Jasaraharja diidentifikasi untuk membuat alur logika dari proses yang berlangsung didalam aplikasi tersebut. Langkah selanjutnya adalah melakukan identifikasi kandidat operasi service yang akan mengenkapsulasi logika dari proses yang berlangsung di dalam aplikasi Dispenda dan Jasaraharja berupa identifikasi kandidat service yang akan dikembangkan.

Mengimplementasikan SOA menggunakan *web service* merupakan hal tepat, karena aplikasi *web service* dapat mewakili sebuah *service* dalam SOA. WSDL dalam arsitektur *web service* dapat menghubungkan satu

service dengan yang lainnya, menggunakan SOAP sebagai teknologi pengiriman pesan antar *service*.

Database menggunakan *MySQL* dan *Ms.Access* yang secara langsung sudah terpasang didalam perangkat Berbasis Web selanjutnya pengguna dapat menambah, dan memperbaharui database yang sudah ada. Aplikasi berjalan dengan baik pada berbagai *platform* seperti: *Windows*, dan *linux*.

a. Analisa Service Oriented

Untuk melakukan analisa berorientasi terhadap pengembangan sistem informasi Dispenda dan Jasaraharja, dilakukan identifikasi proses bisnis yang melibatkan kedua instansi tersebut. Berikutnya, dilakukan identifikasi entitas yang merupakan kandidat pada operasi *service*.

1. Identifikasi Kebutuhan Aplikasi

Langkah pertama yang harus dilakukan di dalam melakukan analisa *service* adalah mendefinisikan statement kebutuhan dari bisnis (*business requirements*). Salah satu cara mendefinisikan kebutuhan dari bisnis dilakukan dengan pemodelan alur kerja (*workflow*). Pemodelan alur kerja dilakukan dengan memodelkan proses – proses yang terlibat di dalam aplikasi Dispenda dan Jasaraharja pada umumnya.

Pada tahap ini, pemodelan proses dilakukan berdasarkan studi literatur dan observasi mengenai aplikasi Dispenda dan Jasaraharja. Pemodelan proses bisnis dilakukan untuk menentukan lingkup pengembangan dan sebagai titik awal untuk melakukan *service-oriented analysis and design* (SOAD).

Pemodelan proses bisnis dilakukan dengan menggunakan notasi *Workflow Modeling*. Pada tahap awal pemodelan dilakukan pemodelan *business use case* menggunakan notasi UML untuk menggambarkan fungsionalitas dan lingkup aplikasi secara keseluruhan. Model UML dapat berguna untuk memberikan gambaran mengenai interaksi (*user*)

dengan aplikasi, memberikan dasar untuk melakukan identifikasi kandidat *service*. Fungsionalitas aplikasi Dispenda dan Jasaraharja.

2. Identifikasi Kandidat *Business Service*

Langkah ini bertujuan untuk melakukan identifikasi dari langkah – langkah yang terdapat pada proses bisnis yang fungsi logikanya dapat dienkapsulasi ke dalam kandidat operasi dari business service. Kandidat operasi service pada proses bisnis yang dapat dikategorikan sebagai proses logik yang potensial untuk dienkapsulasi oleh kandidat service ditentukan oleh 2 kriteria :

- 1) Langkah tersebut tidak termasuk langkah yang dilakukan secara manual dan tidak dapat atau tidak boleh diautomatisasi.
- 2) Langkah tersebut sudah dilakukan oleh sistem yang sudah ada (*legacy system*)

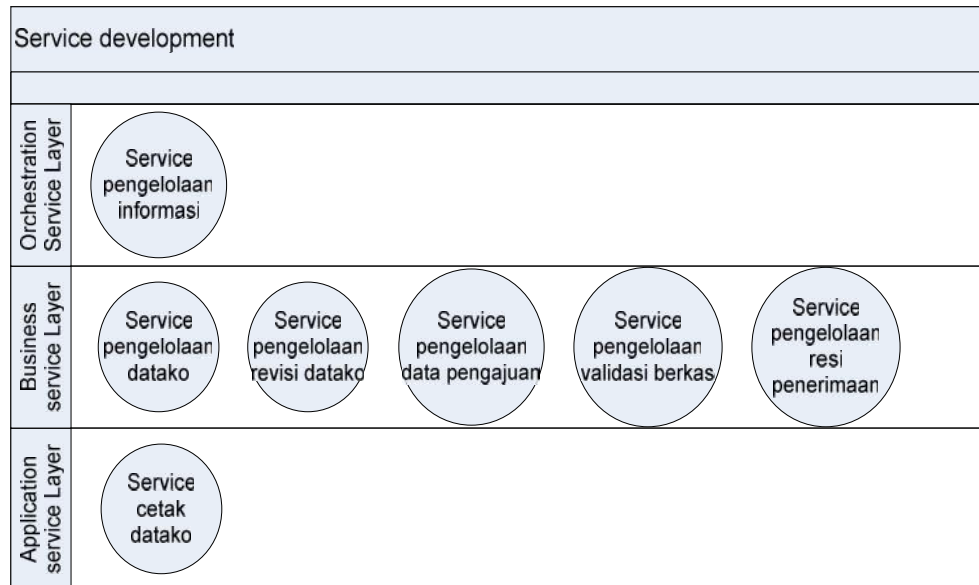
Berdasarkan identifikasi proses bisnis maka di dapat Pemetaan kandidat *Service* seperti yang disajikan pada tabel kandidat service berikut :

Tabel 4.4 Hasil Kandidat *Service*

| No. | Proses Bisnis | Kandidat Service |
|-----|---------------------------------|---|
| 1. | Permintaan ke Dispenda | a. <i>Service</i> pengelolaan datako |
| 2. | <i>Update</i> Datako | a. <i>Service</i> pengelolaan datako b. <i>Service</i> revisi datako |
| 3. | Permintaan ke Sub Bagian Teknik | a. <i>Service</i> pengelolaan datako |
| 4. | Registrasi pengajuan klaim | a. <i>Service</i> pengelolaan data pengajuan |
| 5. | Validasi berkas | a. <i>Service</i> pengelolaan data pengajuan b. <i>Service</i> pengelolaan validasi berkas |

| | | |
|----|------------------------|---|
| | | c. <i>Service</i> pengelolaan informasi |
| 6. | Update resi penerimaan | a. <i>Service</i> mengelola resi penerimaan |

Dari perspektif teknologi informasi, selanjutnya didapat *service interface layer* yang memiliki 3 layer abstraksi yaitu : *Orchestration Service Layer*, *Business Service Layer*, dan *Application Service Layer*.



Gambar 4.2 Hasil Seleksi Kandidat *Service*

Service interface layer pada Gambar 4.2 menjelaskan suatu abstraksi dari logik bisnis, aplikasi, dan aliran proses secara keseluruhan. *Service interface layer* memiliki 3 layer abstraksi yaitu :

1) *Aplication Service Layer*

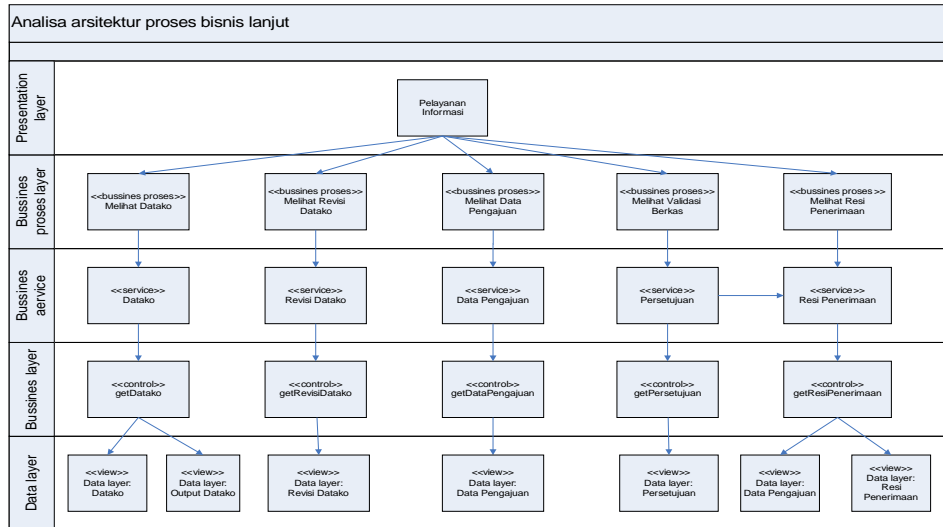
Service yang berada pada layer ini adalah : *service* cetak datako, *service* cetak data pengajuan, dan *service* cetak resi penerimaan.

2) *Business Service Layer*

Service yang terdapat pada layer ini adalah : *service* pengelolaan datako, *service* pengelolaan revisi datako, *service* pengelolaan data pengajuan, *service* pengeloaan validasi berkas, dan *service* pengelolaan resi penerimaan.

3) *Orchestration Service Layer*

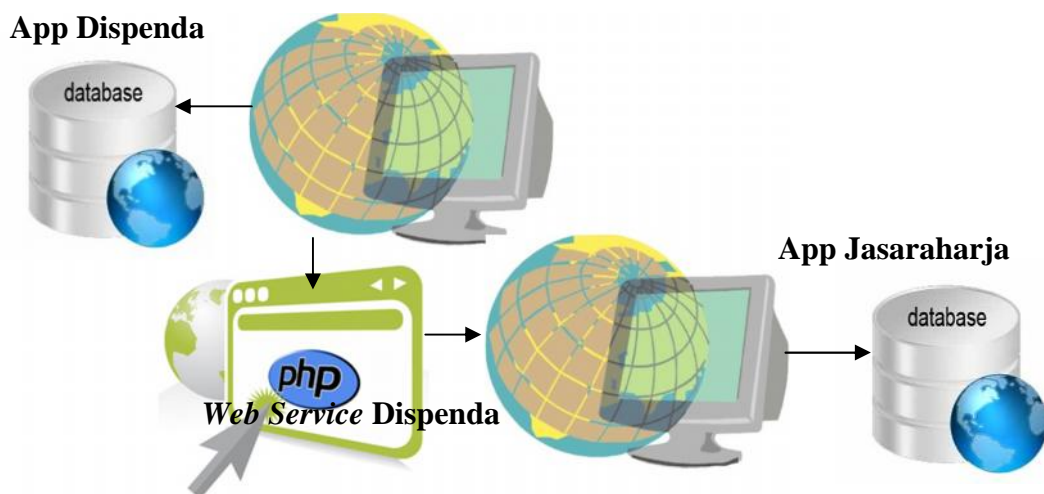
Service yang terdapat pada layer ini adalah *service* pengelolaan informasi.



Gambar 4.3 Arsitektur Sistem Lanjut

3. Dekomposisi Proses Bisnis

Proses yang terdapat pada aplikasi Dispenda dan Jasaraharja berfokus pada 2 proses utama yaitu: proses pengaksesan datako dan proses pengaksesan resi penerimaan. Proses – proses lainnya, seperti proses pengelolaan hak akses user, merupakan pendukung dalam aplikasi yang merupakan proses bisnis dalam mendapatkan identifikasi mengenai kandidat operasi service bisnis. Pada proses bisnis, terdapat 3 aktor utama yaitu: Customer Service di Dispenda, Sub Bagian Teknik, dan Sub Bagian Pelayanan dan Klaim di Jasaraharja.



Gambar 4.4 Identifikasi Arsitektur Sistem

Database yang dibuat adalah *Customer Service* yang didalamnya terdapat tabel bernama *datako*. Tabel *datako* berfungsi untuk menyimpan informasi-informasi seperti: Kode BM, Nama Pemilik, Alamat, Merk, Jenis, Tahun Buat, Tahun Rakit, Isi Silinder, Warna Kendaraan Bermotor, Nomor Rangka, Nomor Mesin, Nomor BPKB, Kode Bahan Bakar, Warna Tanda Nomor Kendaraan Bermotor, Nomor SKPD, Tanggal Daftar, Tanggal Berlaku, Jumlah Pajak Kendaraan Bermotor, Jumlah Sumbangan Wajib, dan Jumlah Total. Sebagai primary key digunakan field Kode BM.

4. Penggambaran *web service*

Dalam penelitian ini *web service* menggunakan sebuah file *xml* yang sudah ada pada library *nusoap*. Terdapat beberapa parameter dalam file tersebut. Berikut adalah potongan algoritma berdasarkan parameter yang digunakan dalam *web service*:

Potongan Algoritma

```
function character_data($parser, $data){
    $pos = $this->depth_array[$this->depth];
    if ($this->xml_encoding=='UTF-8'){
        if($this->decode_utf8){
            $data = utf8_decode($data);
        }
    }
    if (class_exists('SOAPSecurity'))
        $this->message[$pos]['cdata'] .=
            $this->decrypt($data);
    else
        $this->message[$pos]['cdata'] .= $data;
    if($this->status == 'header'){
        $this->responseHeaders .= $data;
    } else {
        $this->document .= $data;
    }
}
```

Gambar 4.5 Potongan Algoritma *Web Service*

4.2 Perancangan Aplikasi

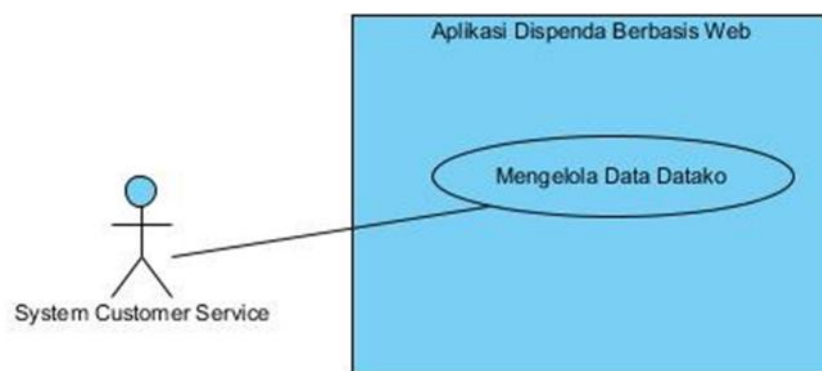
Setelah dilakukan beberapa tahapan dalam analisa sistem, maka dapat dilakukan beberapa perancangan aplikasi *web service* pada perangkat berbasis Web. Perancangan-perancangan yang dijelaskan dalam laporan ini meliputi perancangan model dalam bentuk UML (*Unified Modeling Language*) yang terdiri dari *Usecase Diagram*, *Usecase Diagram Spesification*, *Sequence Diagram*, *Class Diagram*, *Activity Diagram*, dan *Deployment Diagram*. Selain itu juga ada perancangan hubungan antar *class* dengan modul penghubung dan perancangan *interface* sistem yang terdiri dari perancangan *prototype* dan struktur menu.

A. Pemodelan UML (*Unified Modeling Language*)

Analisa perancangan dalam sistem ini menggunakan UML, yaitu : *usecase diagram*, *usecase diagram spesification*, *sequence diagram*, *class diagram*, *activity diagram*, dan *deployment diagram*.

A.1 *Usecase Diagram*

Usecase diagram merupakan suatu aktivitas yang menggambarkan urutan interaksi antar satu atau lebih aktor pada sistem. *Usecase* yang akan dirancang yaitu *usecase diagram* untuk pengaksesan melalui perangkat berbasis Web.



Gambar 4.6 *Usecase Diagram* Dispenda

A.2 Usecase Diagram Spesification

Pada rancangan sistem yang akan dibuat, terdapat aktor dan terdapat *usecase*. Aktor secara langsung berinteraksi dengan sistem, melalui fungsi yang dimiliki pada sistem. Untuk menjelaskan *usecase diagram* pada Aplikasi Berbasis Web dapat dilihat lebih jelas pada *usecase diagram specification*.

1) Mengelola Data Datako

Tabel 4.5 adalah tabel *usecase specification* Mengelola Data Datako yang dilakukan oleh aktor *Customer Service* pada *usecase diagram*.

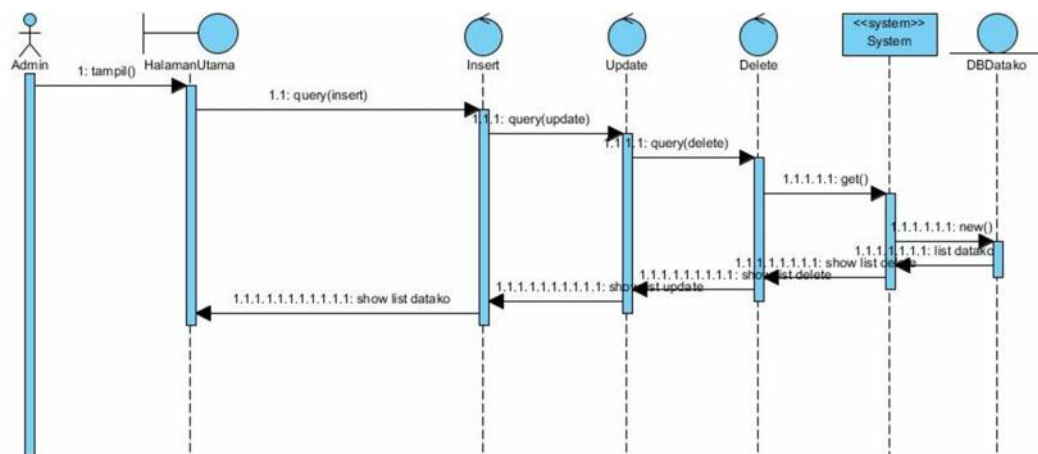
Tabel 4.5 Usecase Spesification Mengelola Data Datako

| | |
|-----------------------|--|
| <i>Usecase</i> | Mengelola Data Datako |
| <i>Description</i> | Digunakan oleh aktor <i>Customer Service</i> untuk Mengelola Data Datako yaitu: <i>insert</i> , <i>update</i> , dan <i>delete</i> kedalam sistem. |
| <i>Actor</i> | <i>Customer Service</i> , Sub Bagian Teknik |
| <i>Pre Condition</i> | Datako belum tersimpan |
| <i>Post Condition</i> | Datako telah tersimpan |
| <i>Normal Flow</i> | <ol style="list-style-type: none">1. Use case dimulai ketika ketika <i>Customer Service</i> ingin melakukan penginputan Datako2. <i>Customer Service</i> memilih menu <i>entry</i> dan memilih menu Kelola Data pada sub menu3. Pada halaman Kelola Data, <i>Customer Service</i> mengisi form Data Pembayaran Pajak4. <i>Customer Service</i> menginputkan Data Pembayaran Pajak secara lengkap5. Data Pembayaran Pajak telah tersimpan6. Datako telah tersimpan7. Sub Bagian Teknik request Kode BM8. Sub Bagian Teknik memilih tombol identifikasi untuk |

| | |
|---------------------------|---|
| | mengidentifikasi Kode BM yang didapat dari <i>web service</i> . |
| <i>Alternate Scenario</i> | a. Setiap field pada form harus diisi, jika tidak sistem akan menolak menyimpan data tersebut. b. Jika data yang sama sudah diisi sebelumnya, maka sistem akan menolaknya. |

A.3 Sequence Diagram

Untuk melakukan login, admin harus memasukkan username dan password dengan benar, jika tidak benar maka akan muncul informasi berhasil atau tidak. Jika login berhasil maka tampilan aplikasi akan muncul. Gambar 4.7 menjelaskan *sequence diagram* untuk mengelola data datako pada perangkat berbasis Web.

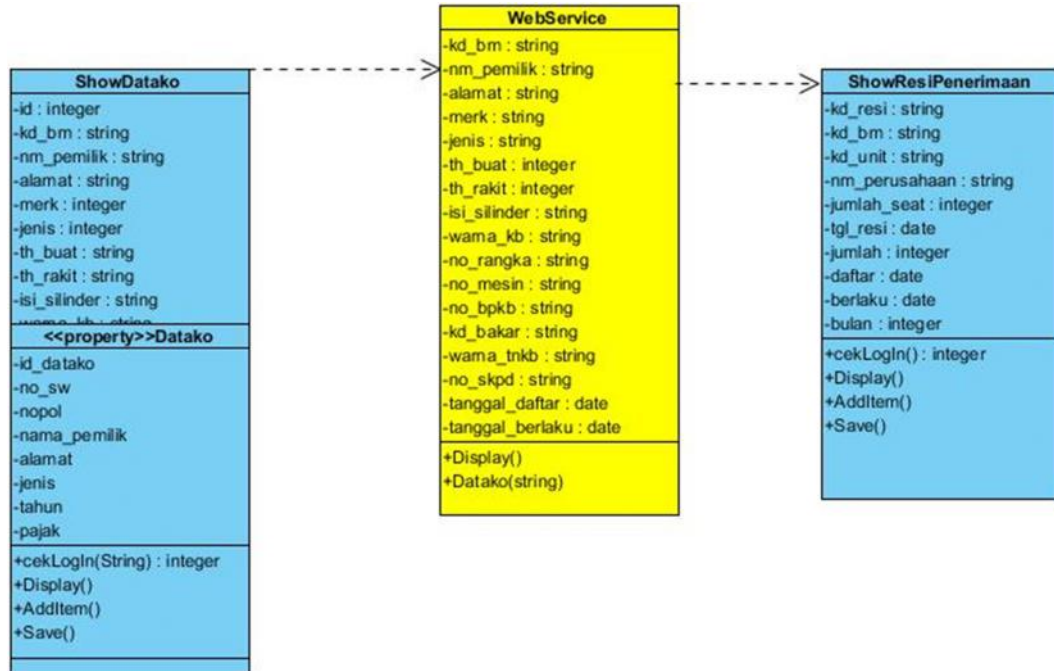


Gambar 4.7 Sequence Diagram untuk Mengelola Data Datako

Dalam hal ini admin Mengelola Data Datako sehingga data akan disimpan dalam database.

A.4 Class Diagram

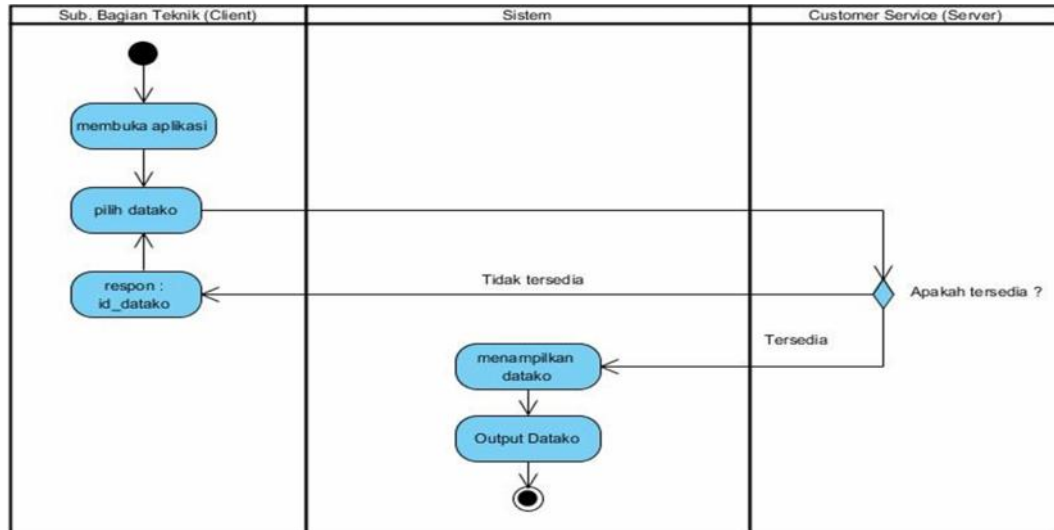
Class diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. *Class diagram* yang dijelaskan pada analisa ini adalah *class diagram* aplikasi yang terpasang pada perangkat berbasis Web.



Gambar 4.8 Class Diagram








A.5 Activity Diagram

Activity diagram merupakan alur kerja pada setiap *usecase*. *Activity diagram* pada analisa ini mencakup *activity diagram* setiap *usecase*. Gambar 4.9 menjelaskan *activity diagram* untuk menampilkan datako pada perangkat berbasis Web.



Gambar 4.9 Activity Diagram Datako

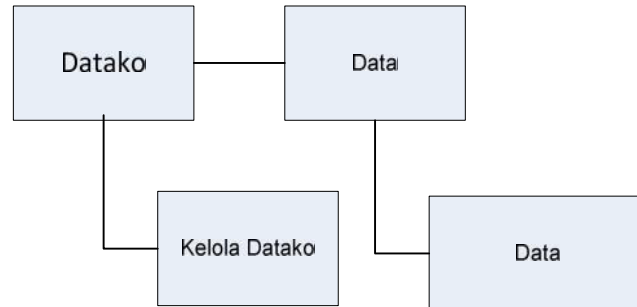
Tabel 4.6 Keterangan Activity Diagram Datako

| Name | Documentation |
|--|--|
|  Swimlane | |
|  Customer Service | |
|  Sub. Bagian Teknik | |
|  InitialNode | |
|  menampilkan datako | Tampilkan datako |
|  Output Datako | Pilihan datako yang di pilih oleh admin yang telah di eksekusi |
|  ActivityFinalNode | |

4.3 Perancangan Menu

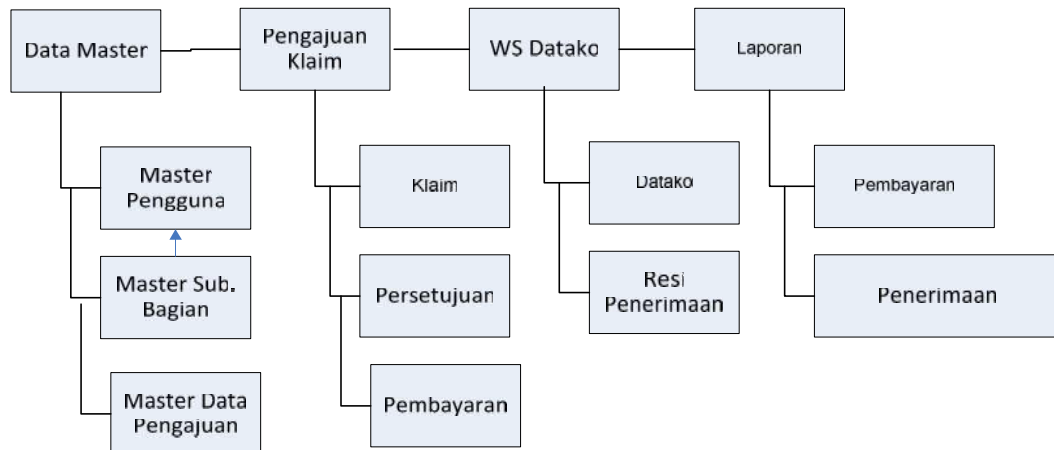
A. Perancangan Struktur Menu Aplikasi

Perancangan struktur menu merupakan tahapan untuk merancang bagaimana struktur menu yang akan dibangun. Berikut struktur menu dari aplikasi yang akan dibangun pada *Customer Service Dispenda* dapat dilihat pada Gambar 4.10.



Gambar 4.10 Perancangan Struktur Menu *Customer Service Dispenda*

Berikut struktur menu dari aplikasi yang akan dibangun pada Sub. Bagian Pelayanan Jasaraharja dapat dilihat pada Gambar 4.11.



Gambar 4.11 Perancangan Struktur Menu Sub. Bagian Pelayanan Jasaraharja

4.4 Perancangan Database

Database merupakan bagian penting dalam aplikasi pada perangkat berbasis Web ini. Seluruh fitur yang ada menggunakan *database* untuk melakukan fungsinya. Perancangan *database* pada aplikasi ini meliputi perancangan *database Customer Service* Dispenda, dan *database Sub. Bagian Pelayanan Jasaraharja*. *Database Customer Service* adalah *database* utama yang ada di Dispenda, dan *Database Sub. Bagian Pelayanan* merupakan *database* pendukung yang di simpan di Jasaraharja.

Berikut adalah Model *Logic Database Customer Service*.

Tabel 4.7 Keterangan Atribut Tabel Datako

| No. | Field | Type | Panjang | Allow Null | Keterangan |
|-----|-----------------|---------|---------|------------|-------------|
| 1. | Kd_bm | Varchar | 30 | Not Null | Primary Key |
| 2. | Nm_pemilik | Varchar | 30 | Not Null | |
| 3. | Alamat | Varchar | 30 | Not Null | |
| 4. | Merk | Varchar | 30 | Not Null | |
| 5. | Jenis | Varchar | 30 | Not Null | |
| 6. | Th_buat | Integer | 11 | Not Null | |
| 7. | Th_rakit | Integer | 11 | Not Null | |
| 8. | Isi_silinder | Varchar | 30 | Not Null | |
| 9. | Warna_kb | Varchar | 30 | Not Null | |
| 10. | No_rangka | Varchar | 30 | Not Null | |
| 11. | No_mesin | Varchar | 30 | Not Null | |
| 12. | No_bpkb | Varchar | 30 | Not Null | |
| 13. | Kd_bakar | Varchar | 30 | Not Null | |
| 14. | Warna_tnkb | Varchar | 30 | Not Null | |
| 15. | No_skpd | Varchar | 30 | Not Null | |
| 16. | Tanggal_daftar | Date | Time | Not Null | |
| 17. | Tanggal_berlaku | Date | Time | Not Null | |
| 18. | Jumlah_pkb | Integer | 11 | Not Null | |
| 19. | Jumlah_sw | Integer | 11 | Not Null | |

| | | | | | |
|-----|--------------|---------|----|----------|--|
| 20. | Jumlah_total | Integer | 11 | Not Null | |
|-----|--------------|---------|----|----------|--|

Untuk Model *Logic Database* Sub. Bagian Pelayanan.

Tabel 4.8 Keterangan Atribut Tabel Resi Penerimaan

| No. | Field | Type | Panjang | Allow Null | Keterangan |
|-----|---------------|---------|---------|------------|-------------|
| 1. | Kd_resi | Varchar | 30 | Not Null | Primary Key |
| 2. | Kd_bm | Varchar | 30 | Not Null | |
| 3. | Kd_unit | Varchar | 30 | Not Null | |
| 4. | Nm_perusahaan | Varchar | 30 | Not Null | |
| 5. | Jumlah_seat | Integer | 11 | Not Null | |
| 6. | Tgl_resi | Date | Time | Not Null | |
| 7. | Jumlah | Integer | 11 | Not Null | |
| 8. | Daftar | Date | Time | Not Null | |
| 9. | Berlaku | Date | Time | Not Null | |
| 10. | Bulan | Integer | 11 | Not Null | |

Untuk detail Model *Logic Database* lainnya yang sedang berjalan di Dispenda, dan PT. Jasaraharja dapat dilihat pada lampiran B.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini merupakan bagian pengkodean aplikasi, dan akan dilakukan implementasi aplikasi dan pengujian fungsi-fungsi aplikasi dengan metode *Blackbox* sebagai bagian dari pengujian aplikasi terhadap pengguna, kemudian akan dilakukan pengamatan dari hasil pengujian tersebut untuk mengetahui kekurangan aplikasi dan kemudian dilakukan pengambilan kesimpulan yang melihat apakah aplikasi yang dibuat sesuai dengan analisa dan perancangan.

5.1 Implementasi

Implementasi *Service Oriented Architecture* (SOA) menggunakan teknologi *web service* yang berjalan pada *platform* berbasis *web* untuk memudahkan proses pengujian aplikasi, sehingga dapat dijalankan tanpa koneksi internet.

Aplikasi *web service* merupakan aplikasi yang berjalan dilingkungan perangkat berbasis Web. Terdapat beberapa keterbatasan yang ditemui pada perangkat ini, sehingga perlu diperhatikan untuk menjadi acuan dalam pengembangan aplikasi ini, yaitu diantaranya:

1. Sumber daya yang terbatas, hingga saat ini perangkat berbasis Web yang banyak beredar memiliki kapasitas memori terbatas. Adapun yang umum digunakan saat ini adalah 512Mb.
2. Tampilan antar muka dengan pengguna sangat berpengaruh terhadap waktu tunggu aplikasi hingga aplikasi benar-benar siap digunakan, semakin banyak komponen yang digunakan akan semakin lama pula waktu tunggu yang dibutuhkan.

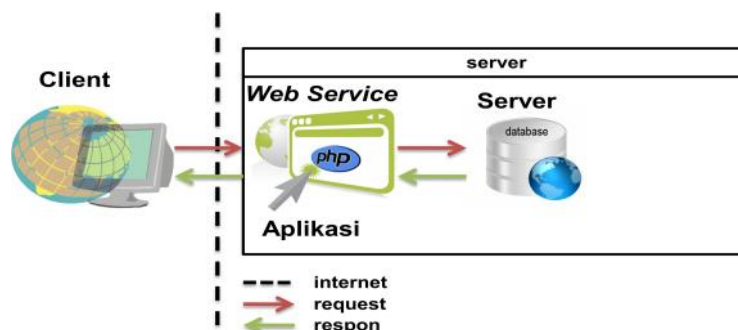
Dari keterbatasan-keterbatasan pada perangkat berbasis Web, maka diusulkan beberapa alternatif untuk meningkatkan performa aplikasi terhadap keterbatasan yang ada, diantaranya:

1. Merancang aplikasi yang menggunakan memori seefektif mungkin, sehingga tidak mengganggu siklus operasi pada aplikasi lain.
2. Merancang aplikasi dengan pemanfaatan sumber daya seefisien mungkin namun tidak mengurangi fungsi dan performa aplikasi.
3. Merancang aplikasi dengan antarmuka yang sederhana namun tetap menarik dan ramah bagi pengguna.

A. Arsitektur Aplikasi

Aplikasi *web service* yang dibahas dalam penelitian ini adalah aplikasi melalui perangkat berbasis Web. Aplikasi ini menggunakan *web service* menggunakan bahasa XML sebagai penghubung dalam pengiriman *request* dan penerimaan respon dari *web service* yang menggunakan Apache dan *database* MySQL.

Aplikasi *web service* ini hanya mampu mengakses data berupa teks dan memberikan akses kepada Sub. Bagian Pelayanan Jasaraharja tanpa bisa memanipulasi *database*. Aplikasi ini memungkinkan Sub. Bagian Pelayanan Jasaraharja untuk menerima data. Aplikasi *web service* yang akan dibangun merupakan aplikasi yang mampu menampilkan data dari *web service* yang meliputi *database* melalui bahasa pemrograman berbasis Web, kemudian *web service* akan memberikan data kepada Sub. Bagian Pelayanan Jasaraharja melalui jalur pengiriman *web service*. Untuk lebih jelasnya deskripsi arsitektur sistem ini dapat di lihat pada Gambar 5.1.



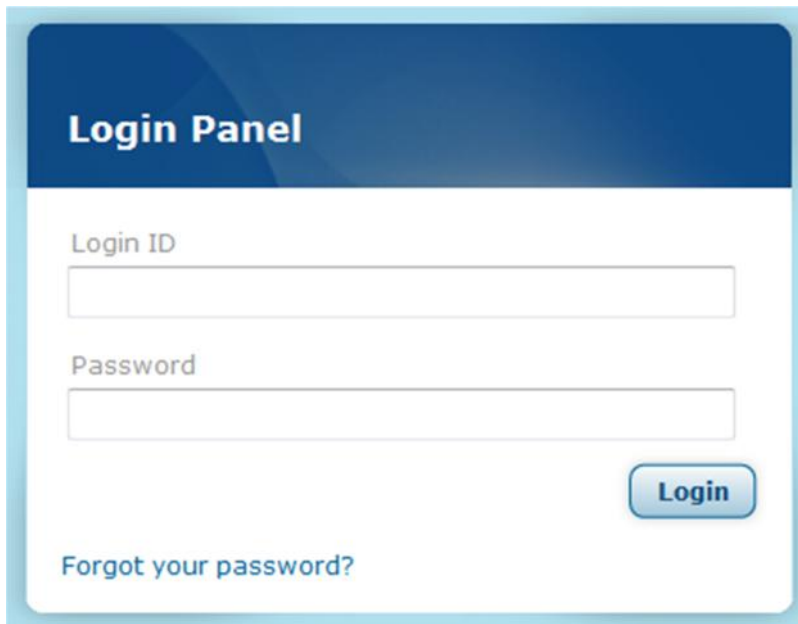
Gambar 5.1 Arsitektur Aplikasi

B. Implementasi *Form Customer Service Dispenda*

Pada tahap-tahap implementasi ini dijelaskan bagaimana tahapan yang penulis lakukan dalam implementasi aplikasi yang telah dibangun.

1) *Form Login*

Form login ini merupakan tampilan pertama kali ketika menjalankan aplikasi. Seorang pengguna harus menginputkan data *login* dengan mengisi Login ID dan *Password* yang tepat sesuai dengan data *login* yang ada di database kemudian menekan tombol *Login* maka pengguna dari sistem ini dapat masuk ke dalam Aplikasi.

The image shows a web form titled "Login Panel" in a dark blue header. Below the header, there are two input fields: "Login ID" and "Password". The "Password" field is masked with dots. To the right of the "Password" field is a blue "Login" button. Below the input fields, there is a link that says "Forgot your password?". The entire form is enclosed in a light blue border.

Gambar 5.2. *Form Login*

Form login ini dibangun menggunakan bahasa pemrograman PHP untuk melakukan proses penyimpanan ke *database* MySQL.

C. Implementasi *Form* Utama Jasaraharja

Pada tahap-tahap implementasi ini dijelaskan bagaimana tahapan yang penulis lakukan dalam implementasi aplikasi yang telah dibangun.

1) *Form Login*

Form ini merupakan tampilan pertama kali ketika menjalankan aplikasi. Seorang pengguna harus menginputkan data *login* dengan mengisi *Username* dan *Password* yang tepat sesuai dengan data *login* yang ada di database kemudian menekan tombol *login* maka pengguna dari sistem ini dapat masuk ke dalam Aplikasi.



Gambar 5.3 *Form Login*

Form login ini dibangun menggunakan bahasa pemrograman PHP untuk melakukan proses penyimpanan ke *database* MySQL.

D. Arsitektur *Web Service*

Dalam arsitektur ini, suatu aplikasi dimodelkan sebagai sekumpulan service. Arsitektur *web service* tersusun dari beberapa komponen yang semuanya berbasis xml (*Extensible Markup Language*) yaitu : SOAP, WSDL, dan UDDI.

1) *Simple Object Access Protocol (SOAP)*

SOAP merupakan suatu protocol berbasis XML yang digunakan untuk kebutuhan pertukaran informasi dalam suatu sistem. *NuSOAP* merupakan library yang dibuat untuk mempermudah proses pembuatan dan pengaksesan *Web Service*.

SOAP

```
<definitions xmlns:SOAP
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs
="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://ww
.w3.org/2001/XMLSchema-instance" xmlns:SOAP
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tn
="urn:DatakoService" xmlns:soap="http://schemas.xmlsoap.
org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl
/" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamesp
ce="urn:DatakoService">
```

Gambar 5.4. SOAP Request

2) *Web Service Description Language (WSDL)*

WSDL menampilkan halaman *web service* yang sudah ada pada perangkat berbasis Web. WSDL memberikan fasilitas untuk pengelolaan *web service* yang dikelola oleh *Customer Service*. *Database* yang dibuat adalah *Customer Service* yang didalamnya terdapat tabel bernama datako.

Tabel datako berfungsi untuk menyimpan informasi-informasi seperti: Kode BM, Nama Pemilik, Alamat, Merk, Jenis, Tahun Buat, Tahun Rakit, Isi Silinder, Warna Kendaraan Bermotor, Nomor Rangka, Nomor Mesin, Nomor BPKB, Kode Bahan Bakar, Warna Tanda Nomor Kendaraan Bermotor, Nomor SKPD, Tanggal Daftar, Tanggal Berlaku, Jumlah Pajak Kendaraan Bermotor, Jumlah Sumbangan Wajib, dan Jumlah Total. Sebagai primary key digunakan field Kode BM.

Selanjutnya dibuat *Web Service* Datako yang menyediakan sebuah fungsi bernama `getDatakoInfo`. Fungsi ini digunakan untuk mengembalikan seluruh daftar Kode BM yang ada pada tabel Datako dalam bentuk *array*. *Web Service* yang dibuat dalam koding terdapat pada Gambar 5.5 berikut.

WSDL

```
<definitions xmlns:SOAP
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xs
="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://ww
.w3.org/2001/XMLSchema-instance" xmlns:SOAP
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tn
="urn:DatakoService" xmlns:soap="http://schemas.xmlsoap.
org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/
" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamesp
ce="urn:DatakoService">
<types>
<xsd:schema targetNamespace="urn:DatakoService">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/e
ncoding/" />
<xsd:import namespace=http://schemas.xmlsoap.org/wsdl/
><xsd:complexType name="datako">
<xsd:all>
<xsd:element name="kd_datako" type="xsd:string" />
<xsd:element name="nm_datako" type="xsd:string" />
<xsd:element name="alamat" type="xsd:string" />
</xsd:all>
</xsd:complexType>
.....
</types>
.....
<message name="getDatakoIdRequest" />
<message name="getDatakoIdResponse">
<part name="return" type="tns:idArray" />
</message>
<message name="getDatakoInfoRequest">
<part name="datakoId" type="xsd:string" />
</message>
<message name="getDatakoInfoResponse">
<part name="return" type="tns:datako" />
</message>
<portType name="DatakoPortType">
<operation name="getDatakoId">
<input message="tns:getDatakoIdRequest" />
<output message="tns:getDatakoIdResponse" />
</operation>
<operation name="getDatakoInfo">
<input message="tns:getDatakoInfoRequest" />
<output message="tns:getDatakoInfoResponse" />
</operation>
</portType>
.....
</port>
</service>
</definitions>
```

Gambar 5.5. Web Service

3) *Universal Description Discovery Integration (UDDI)*

UDDI merupakan sekumpulan spesifikasi yang menunjukkan *registry* informasi mengenai *web service*.



Gambar 5.6. Universal Description Discovery Integration (UDDI)

5.2 Pengujian

Pengujian dilakukan untuk mengetahui seberapa tinggi tingkat keberhasilan aplikasi yang dibuat menggunakan teknologi *web service* yang diimplementasikan pada *Service Oriented Architecture (SOA)* dan melihat apakah aplikasi yang dibuat sesuai dengan analisa dan perancangan.

A. Pengujian *Blackbox*

Pengujian *blackbox* digunakan untuk menguji komponen-komponen yang ada pada aplikasi tanpa memperhatikan struktur logika internalnya. Tahapan pengujian dilakukan untuk mengetahui apakah aplikasi yang dibangun telah sesuai dengan yang diharapkan. Pada tahap pengujian aplikasi ini *platform* yang digunakan berbasis Web. Pengujian yang dilakukan adalah pengujian akses ke sistem dan pengujian fungsionalitas sistem menggunakan metode *blackbox*. Tujuan utama dari pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Salah satu metode pengujian jenis ini dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian *Blackbox*

| No | Komponen yang diuji | Properti Komponen | Hasil Yang Diharapkan | Hasil Pengujian |
|----|----------------------------|--|---|-----------------|
| 1 | <i>Koneksi Web Service</i> | Tombol menu <i>input datako</i> | Menampilkan <i>form input datako</i> | Benar |
| | | Data datako | Menampilkan hasil data <i>datako</i> | Benar |
| | | Tombol menu <i>input resi penerimaan</i> | Menampilkan <i>form input resi penerimaan</i> | Benar |

Untuk pengujian lebih lanjut dapat dilihat pada Lampiran A.

B. Kesimpulan Pengujian

Web service yang disediakan *Customer Service* untuk memberikan layanan datako ke Jasaraharja dapat berjalan dengan baik. Data yang diberikan *web service* berupa revisi datako, pelaporan, dan monitoring. Beberapa aplikasi yang menggunakan database berbeda seperti: *MySQL* dan *Ms. Acces* dapat diintegrasikan menggunakan *web service*. Aplikasi berjalan dengan baik pada berbagai *platform* seperti: *Windows*, dan *linux*.

BAB VI

PENUTUP

6.1 Kesimpulan

Kesimpulan dari implementasi dan pembahasan aplikasi adalah :

1. Dengan adanya aplikasi ini, maka proses pengolahan data akan lebih cepat dan tepat serta akurat dibandingkan dengan sistem pengolahan data manual yang masih digunakan sebelumnya.
2. Teknologi *web service* berjalan dengan baik dari segi aplikasi, untuk mengintegrasikan data dari berbagai aplikasi.
3. Aplikasi ini dapat memberikan informasi dan laporan setiap bulan atau periodenya.
4. Aplikasi berjalan dengan baik pada berbagai *platform*.

6.2 Saran

Saran-saran yang harus diperhatikan untuk penggunaan aplikasi ini :

1. Pada pengembangan aplikasi berikutnya dapat di buat dengan berbasis teknologi keamanan yang lebih baik pada saat pengiriman data, untuk mengoptimalkan aplikasi berjalan secara baik dan lancar.

DAFTAR PUSTAKA

Winarno, ST. Edy. Android, *“Mobile Web Development dengan Dreamweaver”*. Jakarta: PT Alex Media Komputindo. 2010.

Sulistiyani, Sri. Java. *Java for Mobile Programming*. Yogyakarta: CV Andi. 2012.

Jasa Raharja, PT. *Salinan Undang-Undang No. 33 Tahun 1964 Tentang Dana Pertanggunggunaan Wajib Kecelakaan Penumpang*. Jakarta. 2010.

Jasa Raharja PT. *Profil PT. Jasa Raharja (Persero)*. Jakarta. 2010.

Erl, Thomas. *Service-Oriented Architecture: Concepts, Technology and Design*. New York. 2010.

Joko Prasetyo, Hendro. *“Implementasi Service Oriented Architecture (SOA) Menggunakan Teknologi Web Service”*. Fakultas MIPA, Universitas Widya Dharma Klaten. 2009.

Sartono, Arief, B.Sc MM. & Iswahyudi, Annas, S.Kom. *“Rancang Bangun Layanan Berbasis SOA yang Teramankan sebagai Alternatif Solusi Permasalahan DPT pada Pemilu Indonesia”*. Konferensi Teknologi Informasi dan Komunikasi untuk Indonesia Jurnal. Bandung: 2011.

Dear. *“Judul Skripsi/Tugas Akhir Teknik Informatika, Manajemen Informatika, Sistem Informasi, Teknik Komputer, Teknik Elektro”*. Universitas Sriwijaya. 2008.

Frisandi Fachry, dkk. *“Implementasi Service-Oriented Architecture pada Pengembangan Sistem Pembelajaran Mobile”*. Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. 2009.

